



TUGAS AKHIR - KI141502

STUDI KINERJA METODE *LOSSY COMPRESSION* PADA CITRA MEDIS MENGGUNAKAN ALGORITMA KOMPRESI BERBASIS FRAKTAL DAN WAVELET

**NANANG TAUFAN BUDIANSYAH
NRP 5113100183**

**Dosen Pembimbing I
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.**

**Dosen Pembimbing II
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.**

**DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**



TUGAS AKHIR - KI141502

**STUDI KINERJA METODE *LOSSY COMPRESSION* PADA
CITRA MEDIS MENGGUNAKAN ALGORITMA KOMPRESI
BERBASIS FRAKTAL DAN WAVELET**

**NANANG TAUFAN BUDIANSYAH
NRP 5113100183**

**Dosen Pembimbing I
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.**

**Dosen Pembimbing II
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.**

**DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - KI141502

PERFORMANCE STUDY OF LOSSY COMPRESSION METHOD ON MEDICAL IMAGES USING FRACTAL COMPRESSION ALGORITHM AND WAVELET

**NANANG TAUFAN BUDIANSYAH
NRP 5113100183**

**Supervisor I
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.**

**Supervisor II
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2017**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

STUDI KINERJA METODE *LOSSY COMPRESSION* PADA CITRA MEDIS MENGGUNAKAN ALGORITMA KOMPRESI BERBASIS FRAKTAL DAN WAVELET

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh

NANANG TAUFAN BUDIANSYAH
NRP : 5113100183

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.
NIP: 197512202001122002
2. Dr. Eng. Nanik Suciati, S.Kom., M.Kom.
NIP: 197104281994122001



SURABAYA
JUNI, 2017

[Halaman ini sengaja dikosongkan]

STUDI KINERJA METODE *LOSSY COMPRESSION* PADA CITRA MEDIS MENGGUNAKAN ALGORITMA KOMPRESI BERBASIS FRAKTAL DAN WAVELET

Nama Mahasiswa : NANANG TAUFAN BUDIANSYAH
NRP : 5113100183
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.
Dosen Pembimbing 2 : Dr.Eng. Nanik Suciati, S.Kom.,
M.Kom.

Abstrak

Dewasa ini penggunaan komputer semakin meningkat, peningkatan jumlah pengguna komputer mengakibatkan meningkatnya penggunaan data digital. Salah satu data digital yang banyak digunakan yaitu citra digital. Elemen terkecil pada suatu citra digital dinamakan pixel.. Semakin banyak pixel, semakin tinggi resolusi suatu citra digital. Semakin banyak pixel maka semakin besar juga ukuran file citra digital tersebut sehingga akan mengakibatkan kapasitas ruang penyimpanan data.

Untuk memperkecil ukuran file suatu citra digital guna menghemat kapasitas ruang penyimpanan, maka digunakanlah teknik kompresi terhadap citra digital. Metode kompresi citra digital dibagi menjadi 2, yaitu lossy dan lossless data compression. Lossy data compression merupakan metode kompresi data yang menghilangkan beberapa data pada citra digital sehingga menyebabkan pada saat proses dekompresi data maka cira yang dihasilkan berbeda dengan citra yang asli. Terdapat beberapa algoritma dan transformasi untuk kompresi lossy, diantaranya seperti fourier, cosinus, walsh, hamadard, fraktal, dan wavelet. Tugas Akhir ini menggunakan metode

fraktal dan wavelet untuk melakukan kompresi pada citra medis.

Dari hasil uji coba menunjukan bahwa dengan metode wavelet menghasilkan nilai PSNR tertinggi mencapai 33.33. dan untuk metode kompresi fraktal menghasilkan PSNR tertinggi mencapai 22.62.

Kata kunci : Citra Digital, Kompresi, Fraktal, Wavelet, PSNR, Lossy

PERFORMANCE STUDIES OF *LOSSY COMPRESSION* ON MEDICAL IMAGES USING FRACTAL COMPRESSION ALGORITHM AND WAVELET

Student's Name : NANANG TAUFAN BUDIANSYAH
Student's ID : 5113100183
Department : Teknik Informatika FTIF-ITS
First Advisor : Dr.Eng. Chastine Fatichah, S.Kom.,
M.Kom.
Second Advisor : Dr.Eng. Nanik Suciati, S.Kom.,
M.Kom.

Abstract

Nowadays, users of computer are increase, the increasing number of computer users resulted in increased of digital data user. One of the most currently used in digital data is digital image. The smallest element in a digital image is called a pixel. The more pixels, the higher resolution of a digital image. The more pixels the larger file size of digital image will result in the capacity of the data storage space.

To reduce the file size of a digital image for saving the capacity of storage space, therefore used techniques of compression for digital images. Data compression method is divided into 2 types, lossy and lossless data compression. Lossy data compression is a method of data compression that eliminates some data in digital images so that the result of decompression digital image is different with the original digital image, there are several algorithms and transformations for lossy compression such as fourier, cosine, Walsh, hamadard, fractal, and wavelet. This Final Project uses fractal and wavelet method to compress the medical image.

From the test results show that with the wavelet method produces the highest PSNR value reached 31.55. And for the fractal compression method get the highest value 22.62.

Keywords: Digital Image, Compression, Fractal, Wavelet, PSNR, Lossy

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“STUDI KINERJA METODE LOSSY COMPRESSION PADA CITRA MEDIS MENGGUNAKAN ALGORITMA KOMPRESI BERBASIS FRAKTAL DAN WAVELET”**.

Bagi penulis, pengerjaan Tugas Akhir ini merupakan sebuah pengalaman yang berharga. Selama pengerjaan Tugas Akhir, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menjalani perkuliahan di Teknik Informatika ITS dan Tugas Akhir ini adalah implementasi dari apa yang telah penulis pelajari.

Selesaiannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Keluarga tercinta, Ibu, Papa, Mbak Ayu yang telah memberikan dukungan moral dan material serta do'a yang tak terhingga untuk penulis. Serta selalu memberikan semangat dan motivasi pada penulis dalam mengerjakan Tugas Akhir ini.
3. Ibu Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. selaku pembimbing I yang telah membimbing dan membantu penulis serta memberikan motivasi dalam menyelesaikan Tugas Akhir ini dengan sabar.
4. Ibu Dr.Eng. Nanik Suciati, S.Kom., M.Kom. selaku pembimbing II yang juga telah membantu, membimbing, dan memotivasi kepada penulis dalam mengerjakan Tugas Akhir ini.

5. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Radityo Anggoro, S.Kom., M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya serta staf karyawan Jurusan Teknik Informatika ITS yang telah memberikan bantuan demi kelancaran administrasi penulis selama kuliah.
6. Hari, Zaza, Dhita, Lino, Saddam, Nindi, Lino, Rifqi, Apip, Mooy, Lusi, Dhea, Gian, Budi, Dhanar dan teman-teman lab Management Informasi yang telah memberi semangat serta menjadi teman seperjuangan dalam menyelesaikan Tugas.
7. Faisal Arif, Steven, Rayhan, Iyan, Subhan, Gerald, Ivan, Eko, Glenn, Wahyu, Nabil, serta teman-teman manyar tirtasri yang selalu memberikan semangat untuk penulis .
8. Mas azis, mbak dian, mbak ajeng, mbak-mbak dan mas-mas yang telah membantu penulis.
9. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga, penulis mengharapkan kritik dan saran yang membangun dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2017

DAFTAR ISI

LEMBAR PENGESAHAN.....	Error! Bookmark not defined.
Abstrak.....	vii
Abstract.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL.....	xvii
DAFTAR KODE SUMBER	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	2
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Metodologi	3
1.7 Sistematika Penulisan Laporan Tugas Akhir.....	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 Citra Digital.....	7
2.1.1 Elemen-Elemen Citra Digital	10
2.1.2 Format File	12
2.1.3 Pixel.....	13
2.1.4 Bith Depth	13
2.2 Pengolahan Citra Digital	13
2.3 Kompresi Citra	15
2.4 Klasifikasi Metode Kompresi.....	17
2.5 Metode Fraktal	18
2.6 Transformasi Wavelet	22
2.7 Mean Square Error (MSE).....	23
2.8 <i>Peak Signal to Noise Ratio</i> (PSNR)	24
2.9 <i>Compression Ratio</i> (CR)	24
2.10 <i>Bits Per Pixel</i> (BPP)	25
BAB III PERANCANGAN SISTEM.....	27
3.1 Data	27

3.1.1	Data Masukan.....	27
3.1.2	Data Keluaran.....	28
3.2	Desain Umum Sistem	28
3.2.1	Kompresi Citra dengan Metode Wavelet	29
3.2.2	Dekompresi Citra dengan Metode Wavelet.....	34
3.2.3	Kompresi Citra dengan Metode Fractal.....	35
3.2.4	Dekompresi Citra dengan Metode Fractal.....	38
BAB IV IMPLEMENTASI.....		41
4.1	Lingkungan Implementasi	41
4.2	Implementasi	41
4.2.1	Kompresi dan Dekompresi Citra dengan Fractal.....	42
4.2.2	Kompresi dan Dekompresi Citra dengan Wavelet ...	46
BAB V UJI COBA DAN EVALUASI.....		49
5.1	Lingkungan Pengujian	49
5.2	Data Pengujian.....	50
5.3	Skenario Uji Coba	51
5.3.1	Skenario Uji Coba Kompresi citra grayscale dengan data masukan citra medis.....	52
5.3.2	Skenario Uji Coba Kompresi citra grayscale dengan data masukan citra Lena.	54
5.3.3	Skenario Uji Coba Kompresi citra grayscale dengan data masukan citra Baboon.....	56
5.3.4	Skenario Uji Coba Kompresi citra color dengan data masukan citra medis.	58
5.3.5	Skenario Uji Coba Kompresi citra color dengan data masukan citra Lena.....	60
5.3.6	Skenario Uji Coba Kompresi citra color dengan data masukan citra Baboon.	62
5.4	Analisa Hasil Uji Coba.....	64
BAB VI KESIMPULAN DAN SARAN.....		71
6.1	Kesimpulan.....	71
6.2	Saran	72
Daftar Pustaka.....		73
BIODATA PENULIS.....		75

DAFTAR GAMBAR

Gambar 2. 1 Contoh sampling citra digital [11].....	8
Gambar 2. 2. Citra sebagai matrik[11]	9
Gambar 2. 3 Representasi kurva $f(x,y)$ dari citra[11].....	9
Gambar 2. 4 (a) proses kompresi (b) proses dekompresi [1]....	16
Gambar 2. 5 Ilustrasi kompresi loseless [1].....	17
Gambar 2. 6 Ilustrasi kompresi lossy [1].....	18
Gambar 2. 7 Standar Proses Kompresi Fraktal	19
Gambar 2. 8 Standar Proses Dekompresi Fraktal.....	21
Gambar 3. 1 Contoh Citra Masukan	26
Gambar 3. 2 Diagram Proses Program	27
Gambar 3. 3 (a) Dekomposisi level 1, (b) Dekomposisi level 2, (c) Dekomposisi level 3.....	28
Gambar 3. 4 Proses pengkodean dengan koefisien haar.....	28
Gambar 3. 5 (a) Dekomposisi level 1, (b) Dekomposisi level 2, (c) Dekomposisi level 3[1].....	29
Gambar 3. 6 <i>Flowchart</i> Kompresi Wavelet	33
Gambar 3. 7 <i>Flowchart</i> Dekompresi Wavelet.....	34
Gambar 3. 8 Contoh matrix dari citra asli	36
Gambar 3. 9 Ilustrasi pencarian nilai pasangan blok range dan blok domain	36
Gambar 3. 10 <i>Flowchart</i> Kompresi Fraktal	37
Gambar 3. 11 <i>Flowchart</i> Dekompresi Fraktal	39
Gambar 5. 1 Gambaran hasil proses kompresi Wavelet.....	50
Gambar 5. 2 Gambaran hasil proses kompresi Fraktal	51
Gambar 5. 3 Hasil Kompresi Citra Grayscale Medis dengan Wavelet	53
Gambar 5. 4 Hasil Kompresi Citra Grayscale Medis dengan Fraktal	54
Gambar 5. 5 Hasil Kompresi Citra Grayscale Lena dengan Wavelet	55
Gambar 5. 6 Hasil Kompresi Citra Grayscale Lena dengan Fraktal	56

Gambar 5. 7 Hasil Kompresi Citra Grayscale Baboon dengan Wavelet	57
Gambar 5. 8 Hasil Kompresi Citra Grayscale Baboon dengan Fraktal	58
Gambar 5. 9 Hasil Kompresi Citra Color Medis dengan Wavelet	59
Gambar 5. 10 Hasil Kompresi Citra Color Medis dengan Fraktal	60
Gambar 5. 11 Hasil Kompresi Citra Color Lena dengan Wavelet	61
Gambar 5. 12 Hasil Kompresi Citra Color Lena dengan Fraktal	61
Gambar 5. 13 Hasil Kompresi Citra Color Baboon dengan Wavelet	63
Gambar 5. 14 Hasil Kompresi Citra Color Baboon dengan Fraktal	63
Gambar 5. 15 Grafik Persentase Kompresi	69

DAFTAR TABEL

Tabel 4. 1. Spesifikasi lingkungan implementasi.....	41
Tabel 5. 1 Spesifikasi lingkungan uji coba	49
Tabel 5. 2 Hasil kompresi citra medis dengan Wavelet.....	52
Tabel 5. 3 Hasil kompresi citra medis dengan Fraktal.....	53
Tabel 5. 4 Hasil kompresi citra Lena dengan Wavelet.....	54
Tabel 5. 5 Hasil kompresi citra Lena dengan Fraktal.....	55
Tabel 5. 6 Hasil kompresi citra Baboon dengan Wavelet.....	56
Tabel 5. 7 Hasil kompresi citra Baboon dengan Fraktal.....	57
Tabel 5. 8 Hasil kompresi citra color medis dengan Wavelet....	58
Tabel 5. 9 Hasil kompresi citra color medis dengan Fraktal....	59
Tabel 5. 10 Hasil kompresi citra color Lena dengan Wavelet..	60
Tabel 5. 11 Hasil kompresi citra color Lena dengan Fraktal...	61
Tabel 5. 12 Hasil kompresi citra color Baboon dengan Wavelet	62
Tabel 5. 13 Hasil kompresi citra color Baboon dengan Fraktal	62
Tabel 5. 14 Hasil penyimpanan citra dengan ekstensi PNG, JPG , dan GIF	64
Tabel 5. 15 Perbandingan ukuran kompresi citra.....	66
Tabel 5. 16 Perbandingan Threshold pada Dekomposisi Wavelet LV1.....	67
Tabel 5. 17 Perbandingan Persentase Rasio Kompresi.....	68

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4. 1 Proses pembagian Blok Range	42
Kode Sumber 4. 2 Proses pembagian Blok Domain	43
Kode Sumber 4. 3 Penskalaan Blok Domain	44
Kode Sumber 4. 4 Pencarian Kemiripan antara Blok Range dan Blok Domain	44
Kode Sumber 4. 5 Transformasi Affine blok range dan blok domain	45
Kode Sumber 4. 6 Proses Dekompresi Fraktal.....	46
Kode Sumber 4. 7 Menghitung nilai A,DH DV,DD untuk setiap chanel RGB	46
Kode Sumber 4. 8 Kuantisasi nilai xh, xv, xd	47
Kode Sumber 4. 9 Kuantisasi dengan nilai threshold minimal 20.....	47
Kode Sumber 4. 10 Proses Dekuantisasi.....	48
Kode Sumber 4. 11 Pembentukan Citra dari File Terkompres	48

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini penggunaan komputer semakin meningkat, peningkatan jumlah pengguna komputer mengakibatkan peningkatan penggunaan data digital. Salah satu data digital yang banyak digunakan yaitu citra digital. Adapun salah satu penggunaan dari citra digital yaitu untuk menyimpan berbagai macam hasil citra medis seperti foto kanker, gigi, tulang, darah, dan lain sebagainya. Elemen terkecil pada suatu citra digital dinamakan *pixel* [1]. Semakin banyak *pixel*, semakin tinggi resolusi suatu citra digital (semakin banyak *pixel*) maka semakin besar juga ukuran file (*byte*) citra digital tersebut sehingga akan mengakibatkan kapasitas ruang penyimpanan data (*storage*). Kebanyakan dari citra medis memiliki resolusi yang tinggi sehingga ukuran filenya cukup besar.

Untuk memperkecil ukuran file suatu citra digital agar menghemat kapasitas ruang penyimpanan digunakanlah teknik kompresi terhadap citra digital. Tujuan dari kompresi yaitu mengurangi atau menghilangkan redundansi data dari suatu citra untuk mendapatkan ukuran file citra digital yang lebih kecil tetapi tetap menjaga kualitas citra digital [5]. Metode kompresi data (citra digital) dibagi menjadi 2, yaitu *lossy* dan *lossless data compression*. *Lossy data compression* merupakan metode kompresi data yang menghasilkan kompresi dengan rasio kompresi yang besar dengan cara menghilangkan beberapa data pada citra digital sehingga menyebabkan pada saat proses dekompresi data (citra digital) yang dihasilkan berbeda dengan data (citra digital) yang asli [9], terdapat beberapa algoritma dan transformasi untuk kompresi *lossy* seperti fourier, cosinus, walsh, hamadard, fraktal, dan wavelet[1]. Sedangkan *lossless data compression* merupakan

metode kompresi data yang hasil dekompresinya sama dengan data (citra digital) aslinya [9].

Selanjutnya, penulis akan melakukan analisis perbandingan kompresi citra medis dengan beberapa pengukuran performansi (*evaluation method*). Ukuran performansi hasil kompresi yang digunakan pada tugas akhir ini yaitu: *Mean Square Error* (MSE), *PSNR (Peak Signal to Noise Ratio)*, *Compression Ratio* (CR), dan *Bits Per Pixel* (BPP).

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana kompresi citra medis menggunakan metode berbasis Fraktal?
2. Bagaimana kompresi citra medis menggunakan metode berbasis Wavelet?
3. Bagaimana performansi kompresi citra medis dengan menggunakan metode berbasis Fraktal dan Wavelet?

1.3 Tujuan

Tujuan dari Tugas Akhir ini adalah

1. Menganalisis dan mengimplementasikan metode Fraktal dan Wavelet pada citra medis.
2. Menganalisis performansi hasil kompresi citra medis dengan menggunakan *Mean Square Error* (MSE), *PSNR (Peak Signal to Noise Ratio)*, *Compression Ratio* (CR), dan *Bits Per Pixel* (BPP).

1.4 Manfaat

Adapun manfaat dari tugas akhir ini adalah untuk memperkecil ukuran file suatu citra medis sehingga menghemat kapasitas ruang penyimpanan.

1.5 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Citra yang digunakan adalah citra medis *grayscale* dan *true color* dengan format **bitmap** (.BMP)
2. Simulasi dilakukan menggunakan Matlab R2014a.
3. Ukuran citra maksimal yang digunakan yaitu 512x512.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.
Tahap pertama dalam proses pengerjaan tugas akhir ini adalah menyusun proposal tugas akhir. Pada proposal tugas akhir ini diajukan analisis perbandingan *lossy compression* pada citra medis.
2. Studi literatur
Pada tahap ini, akan dicari studi literatur yang relevan untuk dijadikan referensi dalam pengerjaan tugas akhir. Studi literatur dapat diambil dari buku, internet, ataupun materi dalam suatu mata kuliah yang berhubungan dengan metode kompresi berbasis fraktal dan wavelet.
3. Analisis dan desain perangkat lunak
Pada tahap ini akan dilakukan analisis yang dimulai dari pemilihan citra medis yang akan digunakan, baik itu *grayscale* maupun *true color*, selanjutnya citra input tersebut akan dilakukan kompresi dengan 2 buah metode

(Fraktal dan Wavelet). Kemudian tahap terakhir yaitu evaluasi dengan menggunakan 4 macam *evaluation method* (MSE, PSNR, CR, dan BPP).

4. Implementasi perangkat lunak

Perangkat lunak ini akan dibangun dengan menggunakan bahasa pemrograman dan kaskas bantu Matlab R2014a dengan fungsi yang sudah tersedia di dalamnya.

5. Pengujian dan evaluasi

Aplikasi akan diuji setelah selesai diimplementasikan menggunakan skenario yang sudah dipersiapkan. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Dengan melakukan pengujian dan evaluasi dimaksudkan juga untuk mengevaluasi jalannya program, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

6. Penyusunan buku Tugas Akhir.

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini merupakan laporan secara lengkap mengenai Tugas Akhir yang telah dikerjakan baik dari sisi teori, rancangan, maupun implementasi sehingga memudahkan bagi pembaca dan juga pihak yang ingin mengembangkan lebih lanjut. Sistematika penulisan buku Tugas Akhir secara garis besar antara lain:

Bab I Pendahuluan

Bab ini berisi penjelasan latar belakang, rumusan masalah, batasan masalah dan tujuan pembuatan

Tugas Akhir. Selain itu, metodologi pengerjaan dan sistematika penulisan laporan Tugas Akhir juga terdapat di dalamnya.

Bab II Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

Bab III Perancangan Perangkat Lunak

Bab ini berisi penjelasan tentang rancangan dari sistem yang akan dibangun. Rancangan ini dituliskan dalam bentuk *pseudocode*.

Bab IV Implementasi

Bab ini berisi penjelasan implementasi dari rancangan yang telah dibuat pada bab sebelumnya. Implementasi disajikan dalam bentuk *code* secara keseluruhan disertai dengan penjelasannya.

Bab V Uji Coba Dan Evaluasi

Bab ini berisi penjelasan mengenai data hasil percobaan dan pembahasan mengenai hasil percobaan yang telah dilakukan.

Bab VI Kesimpulan Dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

[Halaman ini sengaja dikosongkan]

BAB II

TINJAUAN PUSTAKA

Dalam bab ini akan dijelaskan mengenai teori-teori yang merupakan dasar dari pembangunan sistem. Selain itu terdapat penjelasan yang menunjang pengerjaan Tugas Akhir ini sehingga dapat memberikan gambaran secara umum sistem yang akan dibangun.

2.1 Citra Digital

Citra digital merupakan sebuah *array* berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu, suatu citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut[1]. Untuk penyimpanan citra digital diperlukan format file citra. Setiap format memiliki karakteristik masing masing[1]. Citra digital adalah gambar dua dimensi yang dapat ditampilkan pada layar monitor komputer sebagai himpunan berhingga (diskrit) nilai digital yang disebut piksel (*picture elements*). Setiap pixel memiliki nilai (value atau number) yang menunjukkan intensitas keabuan pada pixel tersebut. Derajat keabuan dimana merepresentasikan grey level atau kode warna. Kisaran nilai ditentukan oleh bit yang dipakai dan akan menunjukkan resolusi aras abu-abu (grey level resolution)[5].

1 bit –2 warna: [0,1]

4 bit –16 warna: [0,15]

8 bit –256 warna: [0,255]

24 bit –16.777.216 warna (true color)

Kanal Merah -Red (R): [0,255]

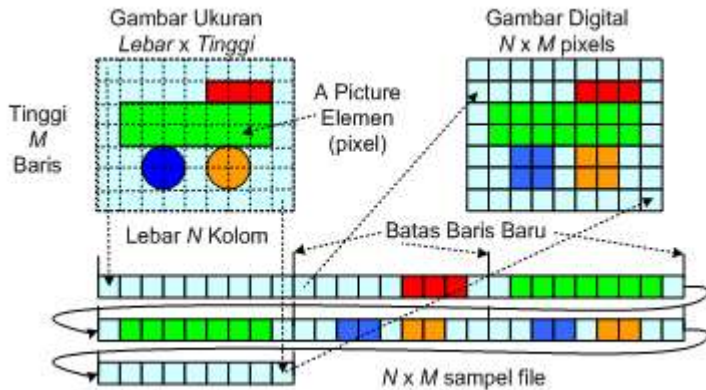
Kanal Hijau - Green (G): [0,255]

Kanal Biru - Blue (B): [0,255]

Suatu citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dinamakan

intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x, y , dan amplitudo f secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital. Gambar 2.2 menunjukkan posisi koordinat digital[11].

Berikut ini adalah contoh sampling dari citra digital :

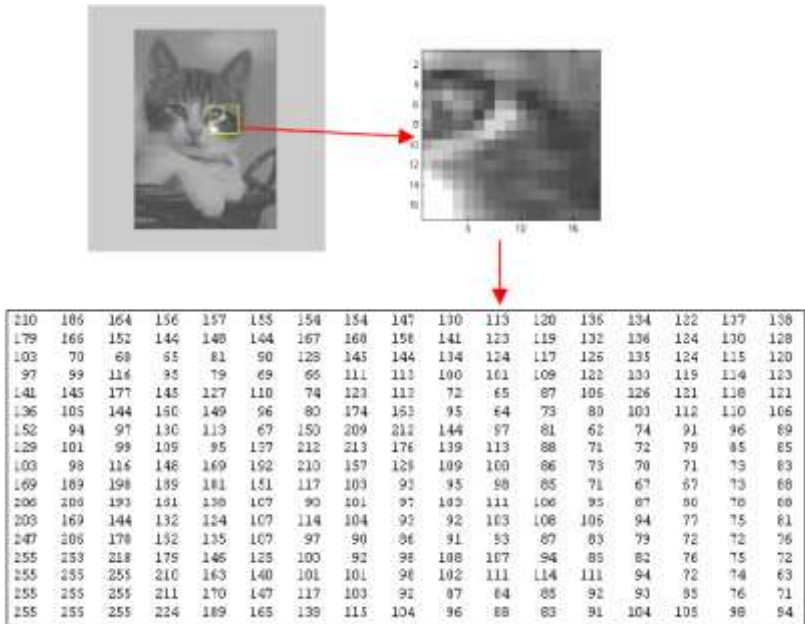


Gambar 2.1 Contoh sampling citra digital [11]

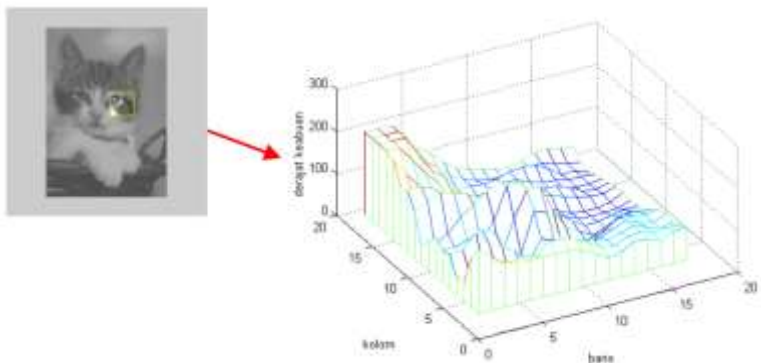
Dengan kata lain, dapat dinyatakan bahwa citra digital adalah citra yang didefinisikan sebagai fungsi $f(x,y)$ di mana x menyatakan nomor baris, y menyatakan nilai kolom, dan f menyatakan nilai derajat keabuan dari citra, dengan demikian (x,y) adalah posisi dari *pixel* dan f adalah nilai derajat keabuan pada titik (x,y) seperti terlihat pada **Gambar 2.2**.

Gambar 2.3 Representasi kurva $f(x,y)$ dari citra[11].

Representasi model di atas adalah representasi dari citra *grayscale* (menggunakan derajat keabuan). Untuk citra berwarna, maka digunakan model *RGB (Red-Green-Blue)* dimana satu citra berwarna dinyatakan sebagai 3 buah matrik *grayscale* yang berupa matrik untuk *Red (R-layer)*, matrik untuk *Green (G-layer)* dan matrik untuk *Blue (B-layer)*.



Gambar 2.2 Citra sebagai matrik[11].



R-layer adalah matrik yang menyatakan derajat kecerahan untuk warna merah misalkan untuk skala keabuan 0-255, maka nilai 0 menyatakan gelap (hitam) dan 255 menyatakan warna

merah. *G-layer* adalah matrik yang menyatakan derajat kecerahan untuk warna hijau, dan *B-layer* adalah matrik yang menyatakan derajat kecerahan untuk warna biru[11].

2.1.1 Elemen-Elemen Citra Digital

Citra digital mengandung sejumlah elemen dasar[5], diantaranya:

1. Kecerahan (*brightness*)

Kecerahan adalah kata lain untuk intensitas cahaya. Kecerahan pada sebuah titik (*pixel*) di dalam citra bukanlah intensitas yang riil, tetapi sebenarnya adalah intensitas rata-rata dari suatu area yang melingkupinya. Sistem visual manusia mampu menyesuaikan dirinya dengan tingkat kecerahan (*brightness level*) mulai dari yang paling rendah sampai yang paling tinggi [5].

2. Kontras (*contrast*)

Kontras menyatakan sebaran terang (*lightness*) dan gelap (*darkness*) di dalam sebuah gambar. Citra dengan kontras rendah dicirikan oleh sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Pada citra dengan kontras baik, komposisi gelap dan terang tersebar secara merata[5].

3. Kontur (*cotour*)

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada *pixel-pixel* yang bertetangga. Karena adanya perubahan intensitas inilah mata kita mampu mendeteksi tepi-tepi (*edge*) objek di dalam citra[5].

4. Warna (*color*)

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang (λ) yang berbeda. Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu (violet) mempunyai panjang gelombang paling rendah.

Warna-warna yang diterima oleh mata (sistem visual manusia) merupakan hasil kombinasi cahaya dengan panjang

gelombang berbeda. Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang warna yang paling lebar adalah *Red* (R), *Green* (G), dan *Blue* (B).

Persepsi sistem visual manusia terhadap warna sangat relatif sebab dipengaruhi oleh banyak kriteria, salah satunya disebabkan oleh adaptasi yang menimbulkan distorsi. Misalnya bercak abu-abu di sekitar warna hijau akan tampak keunguan (distorsi terhadap ruang), atau jika mata melihat warna hijau lalu langsung dengan cepat melihat warna abu-abu, maka mata menangkap kesan warna abu-abu tersebut sebagai warna ungu (distorsi terhadap waktu) [5].

5. Bentuk (*shape*)

Shape adalah *property intrinsic* dari objek tiga dimensi, dengan pengertian bahwa *shape* merupakan *property intrinsic* utama untuk sistem visual manusia. Manusia lebih sering mengasosiasikan objek dengan bentuknya ketimbang elemen lainnya (warna misalnya). Pada umumnya, citra yang dibentuk oleh mata merupakan citra dwimatra (2 dimensi), sedangkan objek yang dilihat umumnya berbentuk trimatra (3 dimensi). Informasi bentuk objek dapat diekstraksi dari citra pada permulaan pra-pengolahan dan segmentasi citra[5].

6. Tekstur (*texture*)

Tekstur dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan *pixel-pixel* yang bertetangga. Jadi, tekstur tidak dapat didefinisikan untuk sebuah *pixel*. Sistem visual manusia pada hakikatnya tidak menerima informasi citra secara independen pada setiap *pixel*, melainkan suatu citra dianggap sebagai suatu kesatuan. Resolusi citra yang diamati ditentukan oleh skala pada mana tekstur tersebut dipersepsi. Sebagai contoh, jika kita mengamati citra lantai berubin dari jarak jauh, maka kita mengamati bahwa tekstur terbentuk oleh penempatan ubin-ubin secara keseluruhan, bukan dari persepsi pola di dalam ubin itu sendiri. Tetapi, jika mengamati citra yang sama dari jarak yang dekat, maka hanya beberapa ubin yang tampak dalam bidang pengamatan,

sehingga kita mempersepsi bahwa tekstur terbentuk oleh penempatan pola-pola rinci yang menyusun tiap ubin[5].

2.1.2 Format File

Format *file* menentukan bagaimana informasi data dipresentasikan dalam suatu *file*. Tiap informasi *file* memiliki kelebihan dan kelemahan pada masing-masing format tersebut. Dalam sistem operasi windows biasanya format *file* dapat dibedakan dari namanya yaitu diakhiri dengan titik misalnya .txt, .doc, .html dan lain-lain. Setiap format memiliki karakteristik masing masing[1]. Berikut ini ini penjelasan beberapa format umum yang digunakan saat ini:

1. Tagged Image Format (.tif, .tiff)

Format .tif merupakan format penyimpanan citra yang dapat digunakan untuk menyimpan citra bitmap hingga citra dengan warna. Format ini dapat digunakan untuk menyimpan citra yang tidak terkompresi dan juga citra terkompresi [1].

2. Portable Network Graphics (.png)

Format .png adalah format penyimpanan data terkompresi. Format ini dapat digunakan pada citra *grayscale*, citra warna, dan juga *fullcolor*[1].

3. JPEG (.jpg)

Format .jpg adalah format yang sangat umum digunakan saat ini khususnya untuk transmisi citra. Format ini juga digunakan untuk menyimpan citra hasil kompresi dengan metode JPEG[1].

4. Graphics Interchange Format (.gif)

Format ini merupakan format yang digunakan pada citra warna 8 bit. Kualitas yang rendah menyebabkan format ini kurang populer di kalangan peneliti pengolahan citra digital[1].

5. Bitmap (.bmp)

Format .bmp adalah format penyimpanan standar tanpa kompresi yang umum dapat digunakan untuk menyimpan citra biner hingga citra warna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan sebuah nilai pixel[1]. Saat ini

format BMP memang “kalah” populer dibandingkan format JPG atau GIF. Hal ini karena berkas BMP pada umumnya tidak dimampatkan, sehingga ukuran berkasnya relatif besar dari pada berkas JPG atau GIF. Hal ini juga yang menyebabkan format BMP sudah jarang digunakan. Meskipun format BMP tidak bagus dari segi ukuran berkas, namun format BMP mempunyai kelebihan dari segi kualitas gambar. Citra dalam format BMP lebih bagus dari pada citra dalam format yang lainnya, karena citra dalam format BMP umumnya tidak dimampatkan sehingga tidak ada informasi yang hilang[5].

2.1.3 Pixel

Pixel disebut juga dengan dot. *Pixel* berbentuk bujur sangkar dengan ukuran relatif kecil yang merupakan penyusun atau pembentuk gambar [11]. Banyaknya *pixel* satuan luas tergantung pada resolusi yang digunakan. Keanekaragaman warna *pixel* tergantung pada *bit depth* yang dipakai. Semakin banyak jumlah *pixel* tiap satuan luas, semakin baik kualitas gambar yang dihasilkan dan tentu akan semakin besar ukuran *filenya*.

2.1.4 Bith Depth

Bit depth (kedalaman warna) yang sering disebut juga dengan *pixel depth* atau *color depth* [11]. *Bit depth* menentukan berapa banyak informasi warna yang tersedia untuk ditampilkan atau dicetak dalam setiap *pixel*. Semakin besar nilainya semakin bagus kualitas gambar yang dihasilkan. Tentu ukuran juga semakin besar.

Misalnya suatu gambar mempunyai *bit depth* = 1. Ini berarti hanya ada 2 kemungkinan warna ($2^1 = 2$) yang ada pada gambar tersebut yaitu hitam dan putih. *Bit depth* = 24 berarti mempunyai warna $2^{24} = 16$ juta warna.

2.2 Pengolahan Citra Digital

Pengolaha citra (*image processing*) merupakan suatu sistem dimana proses dilakukan dengan masukan berupa citra (*image*) dan hasilnya juga berupa citra (*image*) [11].

Macam-macam teknik pengolahan citra digital antara lain sebagai berikut :

1. Perbaikan Kualitas Citra (*Image Enhancement*)
 Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra. Yang termasuk dalam klasifikasi ini antara lain : Perbaikan kontras gelap/ terang (*contrast enhancement*), penajaman (*sharpening*)[11].
2. Pemugaran Citra (*Image Restoration*)
 Operasi ini bertujuan untuk menghilangkan atau meminimumkan cacat pada citra. Yang termasuk dalam klasifikasi ini antara lain : Penghilangan kesamaran (*deblurring*), Penghilangan derau (*noise*)[11].
3. Segmentasi Citra (*Image Segmentation*)
 Operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Jenis operasi ini erat kaitannya dengan pengenalan pola[11].
4. Analisis Citra (*Image Analysis*)
 Operasi ini bertujuan untuk menghitung besaran kuantitatif citra untuk menghasilkan deskripsinya. Yang termasuk dalam klasifikasi ini antara lain : Pendeteksian tepian (*edge detection*), Representasi daerah (*region*), Ekstraksi batas (*boudary*)[11].
5. Kompresi Citra / Pemampatan Citra (*Image Compression*)
 Operasi ini bertujuan untuk memampatkan citra sehingga memori yang dibutuhkan untuk menyimpan citra lebih kecil, tetapi hasil citra yang telah dimampatkan tetap memiliki kualitas gambar yang bagus. Contohnya adalah metode Wavelet, Fractal, Huffman, RLE[11].
6. Rekonstruksi Citra (*Image Reconstruction*)
 Operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis. Contohnya adalah foto

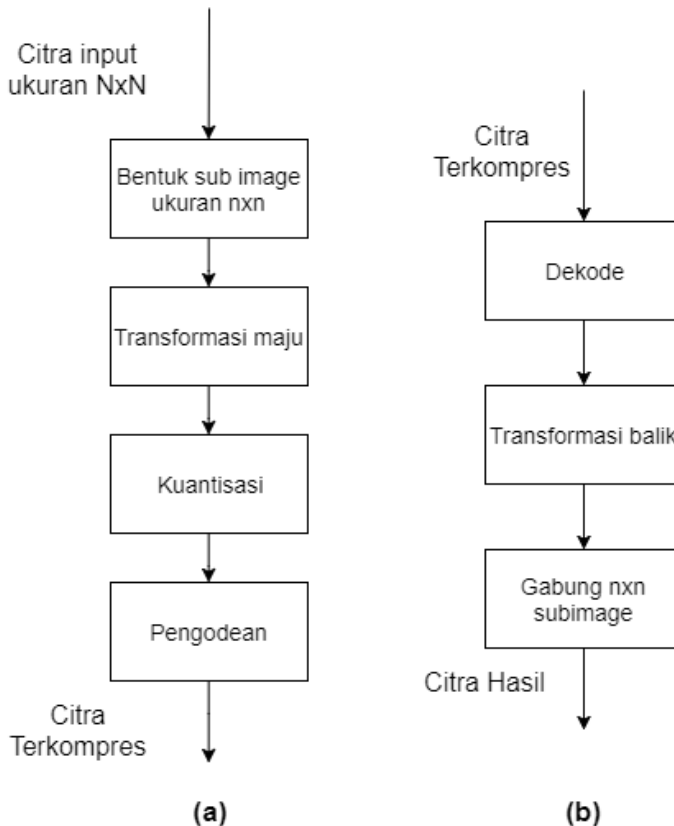
rontgen dengan sinar X digunakan untuk membentuk ulang gambar organ tubuh[11].

Dalam tugas akhir ini, pengolahan citra digital difokuskan pada teknik Kompresi citra.

2.3 Kompresi Citra

Proses kompresi merupakan proses mereduksi ukuran suatu data untuk menghasilkan representasi citra digital yang padat atau mampat namun tetap dapat mewakili kuantitas informasi yang terkandung pada data tersebut. Pada citra, video, dan audio, kompresi mengarah pada minimisasi jumlah bit rate untuk representasi digital. Pada beberapa literatur, istilah kompresi sering disebut juga *source coding*, *data compression*, *bandwidth compression*, dan *signal compression* [8].

Data dan informasi adalah dua hal yang berbeda. Pada data terkandung suatu informasi. Namun tidak semua bagian data terkait dengan informasi tersebut atau pada suatu data terdapat bagian-bagian data yang berulang untuk mewakili informasi yang sama. Bagian data yang tidak terkait atau bagian data yang berulang tersebut disebut dengan data berlebihan (*redundancy data*). Tujuan daripada kompresi data tiada lain adalah untuk mengurangi data berlebihan tersebut sehingga ukuran data menjadi lebih kecil dan lebih ringan dalam proses transmisi. Kompresi citra (*images compression*) mempunyai tujuan meminimalkan kebutuhan memori untuk merepresentasikan sebuah citra digital. Prinsip umum yang digunakan pada proses kompresi citra digital adalah mengurangi duplikasi data di dalam citra sehingga memori yang dibutuhkan untuk merepresentasikan citra menjadi lebih sedikit dari pada citra digital aslinya. Terdapat dua proses utama dalam permasalahan kompresi citra digital [8] seperti yang ditunjukkan pada Gambar 2.4 dibawah.



Gambar 2.4 (a) proses kompresi (b) proses dekompresi [1]

1. Kompresi citra (*images compression*)

Pada proses ini citra digital dalam representasinya yang asli (belum dikompres) dikodekan dengan representasi yang meminimumkan kebutuhan memori. Citra dengan format bitmap pada umumnya tidak dalam bentuk kompresan. Citra yang sudah dikompres disimpan ke dalam arsip dengan menggunakan format tertentu[8]. Kompresi berbasis transformasi citra merupakan kompresi yang bersifat lossy [1].

2. Dekompresi citra (*images decompression*)

Pada proses dekompresi, citra yang sudah dikompresi harus dapat dikembalikan lagi menjadi representasi citra seperti citra aslinya. Proses ini diperlukan jika citra ingin ditampilkan ke layar atau disimpan ke dalam arsip dengan format yang tidak terkompres [1].

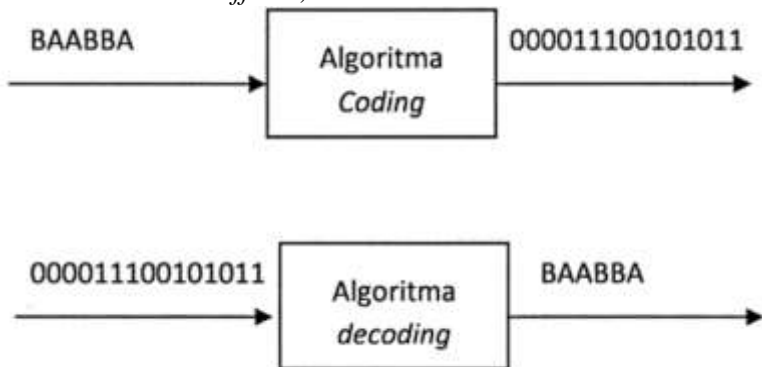
2.4 Klasifikasi Metode Kompresi

Metode kompresi citra dapat diklasifikasikan ke dalam dua kelompok besar :

1. Metode *Loseless*

Metode *loseless* selalu menghasilkan citra hasil pemampatan yang tepat sama dengan citra semula, *pixel per pixel*. Tidak ada informasi yang hilang pada proses pemampatan[1]. Gambaran kompresi *loseless* ditunjukkan pada gambar 2.5 dibawah.

Contoh : metode *Huffman, RLE*.

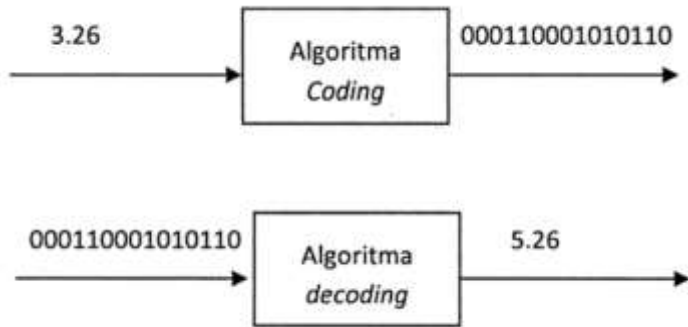


Gambar 2.5 Ilustrasi kompresi *loseless* [1].

2. Metode *Lossy*

Metode *lossy* menghasilkan citra hasil pemampatan yang hampir sama dengan citra semula. Ada informasi yang hilang akibat pemampatan, tetapi dapat ditolerir oleh persepsi mata[1]. Gambaran kompresi *loseless* ditunjukkan pada gambar 2.6 dibawah.

Contoh : Metode *Fractal, Wavelet*

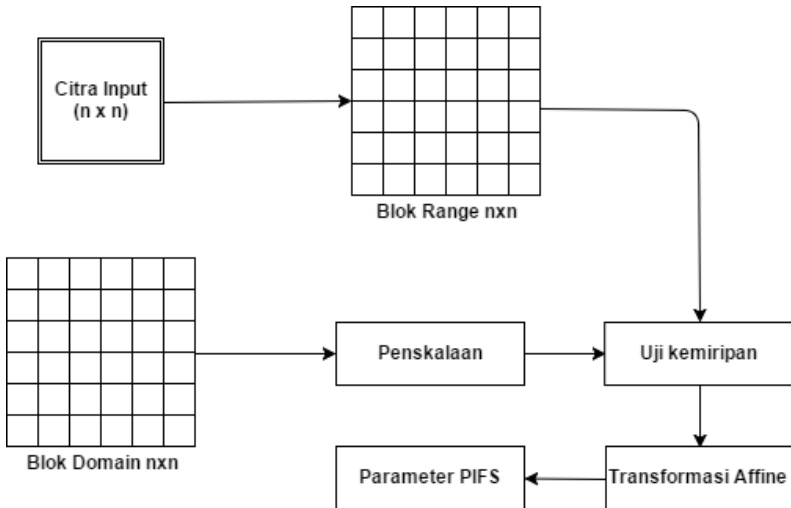


Gambar 2.6 Ilustrasi kompresi lossy [1].

Dalam tugas akhir ini, metode kompresi citra difokuskan pada metode *Lossy* khususnya Fraktal dan Wavelet.

2.5 Metode Fraktal

Metode pemampatan cira fraktal (*fractal image compression*) adalah metode *lossy* compression yang relatif baru. Metode ini mengeksploitasi kemiripan bagian-bagian di dalam citra dan menghitung transformasi yang memetakan bagian-bagian citra yang memiliki kemiripan tersebut[5]. Algoritma Fraktal adalah salah satu dari berbagai metode *lossy image compression* [1]. Metode kompresi ini mengusung fakta bahwa dalam suatu citra ada bagian yang mirip dengan bagian lainnya pada citra tersebut[3]. Kompresi fraktal mengusung fakta bahwa dalam sebuah citra ada suatu bagian yang mirip dengan bagian lain dari citra tersebut. Algoritma fraktal mengubah bagian-bagian yang mirip tersebut menjadi data matematis yang disebut “kode fraktal”, yang akan digunakan untuk membentuk ulang citra yang dikompresi. Citra yang dikompresi dengan metode fraktal akan kehilangan resolusinya, sehingga memungkinkan untuk membentuk kembali citra tersebut dalam resolusi yang berbeda. Gambaran kompresi fraktal terlihat pada gambar 2.7.



Gambar 2.7 Standar Proses Kompresi Fraktal

Adapun algoritma untuk kompresi citra digital dengan fraktal adalah [5] :

- a. Baca citra asli
- b. Menentukan ukuran matriks citra asli
- c. Menentukan ukuran blok *range*
- d. Menentukan ukuran blok domain
- e. Blok domain diskalakan ukurannya menjadi ½ kali ukuran semula.
- f. Untuk setiap blok *range* :
 1. Dicari kemiripan antara blok range ke *i* dengan semua blok domain dengan menghitung rms (root mean square) seperti pada persamaan(2.1). dimana *n* merupakan jumlah pixel pada citra, *z'* merupakan blok range dan *z* merupakan blok domain.

$$d_{rms} = \frac{1}{n} \sqrt{\sum_{i=j}^n \sum_{j=1}^n (z'_{ij} - z_{ij})^2} \quad (2.1)$$

2. Hitung transformasi affine untuk antara blok *range* ke *i* dengan blok domain yang terpilih dengan persamaan (2.2). Dimana Titik (x,y) dengan intensitas z yang termasuk di dalam blok ranah dipetakan oleh w_i , o_i menyatakan offset kecerahan dan parameter s_i adalah kontras, sedangkan e dan f menyatakan pergeseran sudut kiri blok domain ke sudut kiri blok range.

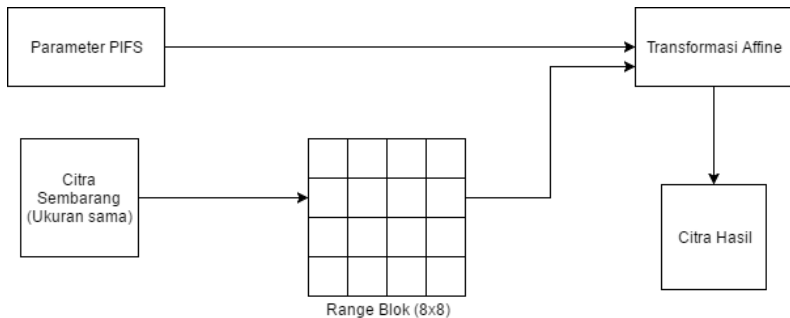
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \quad (2.2)$$

3. Simpan koefisien transformasi affine ke *i* berupa s_i yang menyatakan faktor kontras *pixel* (seperti tombol kontras di TV) yang didapat dengan persamaan (2.3), o_i menyatakan offset kecerahan (*brightness*) *pixel* yang didapat dengan persamaan (2.4), parameter e_i dan f_i yang menyatakan pergeseran sudut kiri domain blok ke sudut kiri range blok yang terpilih. Dimana d merupakan pixel pada blok domain dan r merupakan pixel pada blok range

$$s = \frac{[n \sum_{i=1}^n d_i r_i - \sum_{i=1}^n d_i \sum_{i=1}^n r_i]}{[n \sum_{i=1}^n d_i^2 - (\sum_{i=1}^n d_i)^2]} \quad (2.3)$$

$$o = \frac{1}{n} [\sum_{i=1}^n r_i - s \sum_{i=1}^n d_i] \quad (2.4)$$

4. Simpan semua parameter dalam PIFS.



Gambar 2.8 Standar Proses Dekompresi Fraktal

Proses dekomposisi fraktal ditunjukkan pada Gambar 2.8. Adapun algoritma untuk dekomposisi kompresi citra digital dengan fraktal adalah [5] :

- a. Baca parameter PIFS
- b. Baca Citra sembarang berukuran sama dengan citra input
- c. Membuat Range Blok dari citra sembarang berukuran 8x8.
- d. Untuk setiap Range Blok dilakukan Transformasi Affine dengan parameter PIFS yang ada.
- e. Citra semula terbentuk.

2.5.1 PIFS (Partition Iterated Function System)

Pada kenyataannya sangat sulit menemukan suatu sembarang citra yang *self-similarity* dengan keseluruhan citra. IFS tidak mungkin diterapkan pada sembarang citra. Dengan kata lain IFS hanya dapat berlaku untuk objek-objek Fraktal (objek-objek yang memiliki kemiripan dengan diri sendiri secara sempurna) saja. *Self-similarity* pada sembarang citra dapat diperoleh dengan mencari kemiripan lokal, yaitu mencari bagian-bagian tertentu pada citra yang mirip dengan bagian-bagian lain pada citra tersebut. Metode untuk mencari *self-similarity* lokal ini disebut dengan metode PIFS. Metode diterapkan untuk kompresi citra dan kini telah menjadi algoritma standar dalam pengkodean Fraktal (*Fractal Coding*). Dalam mencari kemiripan lokal suatu citra, PIFS membagi (partisi) citra ke dalam blok-blok besar yang disebut blok

ranah (*domain blok*) dan blok-blok yang lebih kecil yang disebut blok jelajah (*range blok*). Blok ranah boleh saling tumpang tindih sedangkan blok jelajah tidak. Transformasi w_i dinyatakan dengan perluasan transformasi *Affine* pada persamaan [5].

Jadi, titik (x,y) adalah sebuah pixel domain dengan nilai z , dan (x',y') adalah sebuah pixel range dengan nilai baru z' . Koefisien e_i dan f_i mentranslasikan blok *domain* ke dalam posisi blok *range* ketika ukurannya mengecil dalam skala 2 kali.

2.6 Transformasi Wavelet

Wavelet diartikan sebagai *small wave* atau gelombang singkat. Transformasi Wavelet akan mengkonversi suatu sinyal kedalam sederet Wavelet. Gelombang singkat tersebut merupakan fungsi basis yang terletak pada waktu yang berbeda. Wavelet merupakan metode untuk tranformasi sinyal sehingga mampu memberikan informasi frekuensi yang muncul tentang skala atau durasi atau waktu[1]. Transformasi Wavelet merupakan perbaikan dari Transformasi Fourier, Transformasi Wavelet mampu menmbarkan informasi tentang skala atau durasi waktu. Transformasi Wavelet merupakan salah satu metode yang sangat terpakai dan sangat baik digunakan untuk representasi dan analisis sinyal deskret[8]. Ada banyak sekali metode *lossy compression* untuk citra, salah satunya yaitu transformasi wavelet yang juga digunakan pada kompresi gambar standar seperti JPEG dan JPEG2000[7]. Fungsi wavelet dapat didefinisikan dengan Persamaan (2.5) dan (2.6).

$$H_0: y(n) = \frac{1}{2}x(n) + \frac{1}{2}x(n-1) \quad (2.5)$$

$$H_1: y(n) = \frac{1}{2}x(n) - \frac{1}{2}x(n-1) \quad (2.6)$$

Haar merupakan wavelet paling tua dan paling sederhana, diperkenalkan oleh Alfred Haar pada tahun 1909. Haar menjadi sumber ide bagi munculnya keluarga wavelet lainnya[1]. Haar transform untuk kompresi dapat di terapkan pada citra grayscale dan citra warna.

Haar *Function* Persamaan (2.7) dan koefisien haar Persamaan (2.8) :

$$\varphi(t) = \begin{cases} 1, & 0 \leq t < 0.5, \\ -1, & 0.5 \leq t < 1. \end{cases} \quad (2.7)$$

$$\text{Dimana } h_0 = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right) \text{ dan } h_1 = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right) \quad (2.8)$$

Adapun algoritma untuk kompresi citra digital dengan wavelet adalah [1]:

1. Menentukan nilai LL, LH, HL, dan HH dari citra input yang bukur $\frac{1}{4}$ kali ukuran citra aslinya dengan cara memasukkan koefisien seperti pada Persamaan 2.8 sesuai dengan level dekomposisinya.
2. Proses kuantisasi atau menghilangkan koefisien yang tidak dibuthkan
3. Menyimpan hasil kompresi.

2.7 Mean Square Error (MSE)

Mean Square Error merupakan salah satu parameter untuk evaluasi kualitas dari citra yang di kompres[2]. Jika nilai dari MSE kecil, maka kualitas dari citra yang dikompres akan makin baik[2]. Fungsi MSE terdapat pada Persamaan (2.9) berikut:

$$MSE = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N [f(x, y) - f'(x, y)]^2 \quad (2.9)$$

Dimana $f(x,y)$ merupakan citra input asli, $f'(x,y)$ adalah citra yang telah dikompres dan M,N adalah dimensi dari citra.

2.8 *Peak Signal to Noise Ratio (PSNR)*

PSNR merupakan rasio antara ukuran citra input dibagi Mean Square Error (MSE) jika PSNR besar maka kualitas citra akan meningkat [2]. Persamaan (2.10) PSNR adalah sebagai berikut:

$$PSNR = 10 \log_{10} \left[\frac{MXN}{MSE^2} \right] \quad (2.10)$$

Dimana $M \times N$ merupakan ukuran dari citra input.

2.9 *Compression Ratio (CR)*

Compression Ratio merupakan parameter yang sangat berguna untuk kompresi citra[2]. Dengan CR maka dapat ditentukan rasio antara ukuran gambar yang tidak dikompres dengan yang telah dikompres[2]. Fungsi CR dapat dilihat pada Persamaan (2.11) berikut:

$$CR = \frac{UNCOMPRESSED \text{ IMAGE SIZE}}{COMPRESSED \text{ IMAGE SIZE}} \quad (2.11)$$

2.10 *Bits Per Pixel* (BPP)

Bits Per Pixel Ratio merepresentasikan banyaknya bits yang disimpan dalam satu pixel dari citra input[2]. Fungsi BPP dapat dilihat pada Persamaan (2.12) berikut:

$$BPP = \frac{SIZE\ OF\ COMPRESSED\ FILE}{TOTAL\ NO\ OF\ PIXEL\ IN\ THE\ IMAGE} \quad (2.12)$$

[Halaman ini sengaja dikosongkan]

BAB III

PERANCANGAN SISTEM

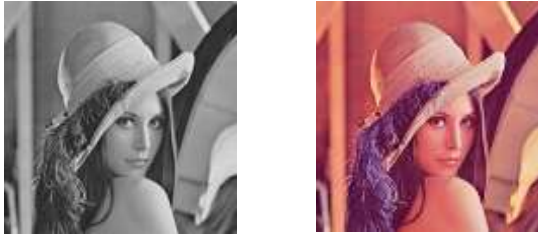
Bab ini menjelaskan tentang perancangan dan pembuatan sistem perangkat lunak. Sistem perangkat lunak yang dibuat pada Tugas Akhir ini adalah mengkompresi data citra masukan dengan metode kompresi citra lossy Wavelet dan Fractal yang selanjutnya akan dilakukan perbandingan terhadap hasil kompresi kedua metode tersebut dari beberapa aspek yaitu ukuran file, waktu kompresi dan hasil citra setelah dikompresi. Pada bab ini pula akan dijelaskan gambaran umum sistem dalam bentuk *flowchart*.

3.1 Data

Pada sub bab ini akan dijelaskan mengenai data yang digunakan sebagai masukan perangkat lunak untuk selanjutnya diolah dan dilakukan perbandingan sehingga menghasilkan data keluaran yang diharapkan.

3.1.1 Data Masukan

Data masukan adalah data yang digunakan sebagai masukan awal dari sistem. Data masukan yang digunakan dalam studi kerja kompresi citra digital dengan metode wavelet dan fractal adalah citra digital *.bmp grayscale dan Warna (RGB) dengan ukuran maksimal 512 x 512 pixel. Contoh data awal yang digunakan pada Tugas Akhir ini terdapat pada Gambar 3.1.



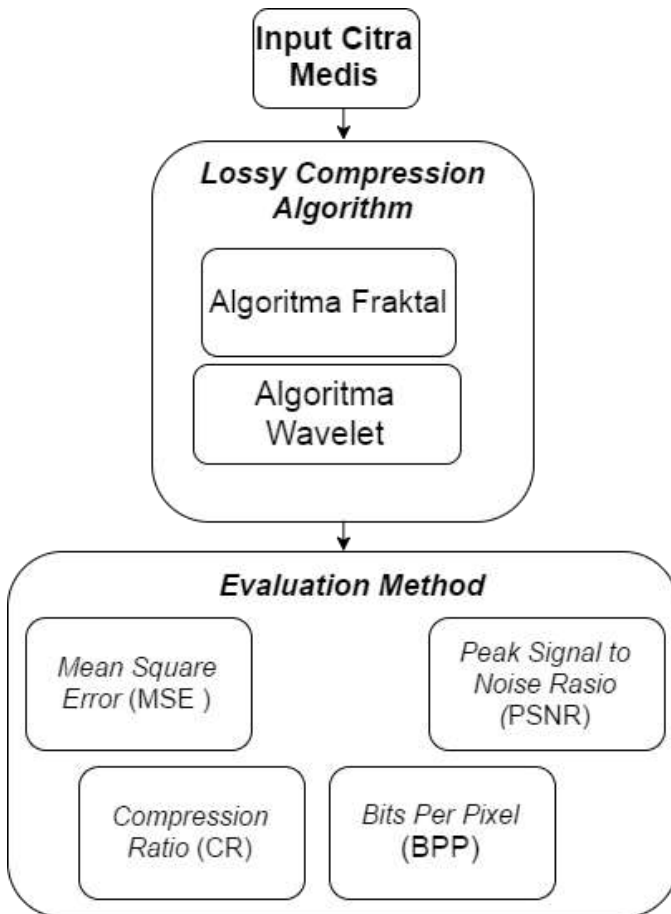
Gambar 3.1 Contoh Citra Masukan

3.1.2 Data Keluaran

Data masukan yang telah dikompresi akan ditampilkan untuk dibandingkan dengan data masukan. Selanjutnya, program yang dikembangkan akan menampilkan informasi hasil kompresi dari citra awal berupa ukuran file dan lama waktu kompresi.

3.2 Desain Umum Sistem

Rancangan bangun perangkat lunak yang akan dibuat untuk studi kerja pemampatan (kompresi) citra digital dengan metode wavelet dan Fractal. Proses pertama yang dilakukan yaitu melakukan kompresi citra masukan dengan metode Wavelet dan Fractal secara bersama – sama, proses kompresi citra bertujuan untuk mendapatkan ukuran citra hasil kompresi dan waktu yang dibutuhkan untuk melakukan masing-masing kompresi citra. Selanjutnya setelah proses kompresi telah selesai dilakukan program akan menampilkan kembali citra hasil dari masing – masing proses kompresi Wavelet dan Fractal yang bertujuan untuk melihat perbandingan kualitas citra sebelum dan setelah dikompresi. Gambar 3.2 menunjukkan alur proses secara umum pada tugas akhir ini.

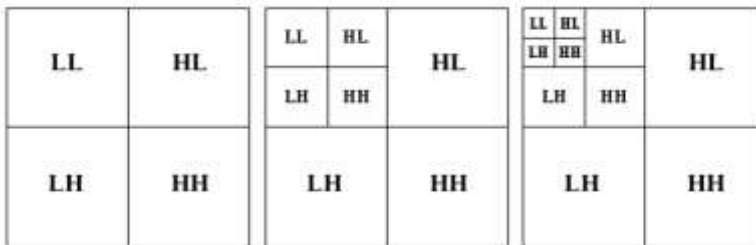


Gambar 3.2 Diagram Proses Program

3.2.1 Kompresi Citra dengan Metode Wavelet

Wavelet dapat diartikan sebagai gelombang singkat. Transformasi Wavelet akan mengkonversi suatu sinyal ke dalam sederet wavelet. Wavelet merupakan metode untuk transformasi sinyal sehingga mampu memberikan informasi frekuensi yang

muncul tentang skala atau durasi atau waktu[1]. Ada banyak sekali metode *lossy compression* untuk citra, salah satunya yaitu transformasi wavelet yang juga digunakan pada kompresi gambar standar seperti JPEG dan JPEG2000[7]. Transformasi wavelet menggunakan dua komponen penting dalam melakukan transformasi yakni fungsi skala (*scaling function*) dan fungsi wavelet (*wavelet function*). Fungsi skala (*scaling function*) disebut juga sebagai *Lowpass Filter*, sedangkan fungsi wavelet (*wavelet function*) disebut juga sebagai *Highpass Filter*. Tahapan diatas disebut juga dengan proses dekomposisi. Hasil dekomposisi digunakan pada saat transformasi wavelet dan inverse transformasi wavelet. Proses dekomposisi dengan metode wavelet ditunjukkan pada Gambar 3.3 dan 3.5.



Gambar 3.3 (a) Dekomposisi level 1, (b) Dekomposisi level 2, (c) Dekomposisi level 3

10	10	20	20
10	10	10	10
50	50	30	30
50	50	30	30

(a)

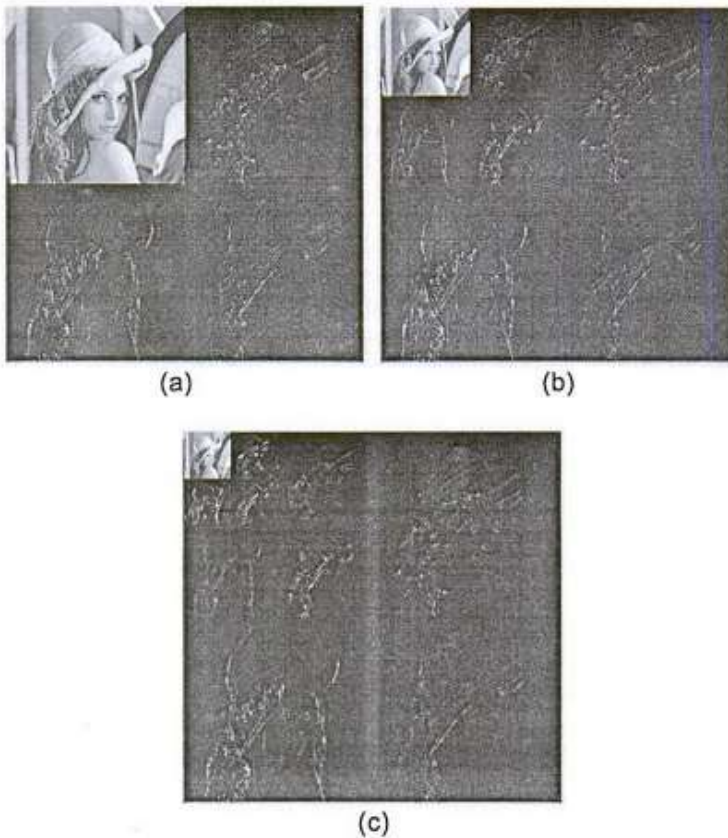
10	10	0	0
10	10	0	0
50	30	0	0
50	30	0	0

(b)

10	15	0	0
50	30	0	0
0	0	0	0
0	0	0	0

(c)

Gambar 3.4 Proses pengkodean (a) citra asli, (b) perhitungan baris, (c) perhitungan kolom



Gambar 3.5 (a) Dekomposisi level 1, (b) Dekomposisi level 2, (c) Dekomposisi level 3[1]

Pada Gambar 3.4 menunjukkan proses pertama dari kompresi wavelet yaitu pengkodean dengan koefisien haar. Sebagai contoh Gambar 3.4 (a) merupakan citra yang memiliki nilai pixel seperti pada gambar. Gambar 3.4 (b) diperoleh dari :

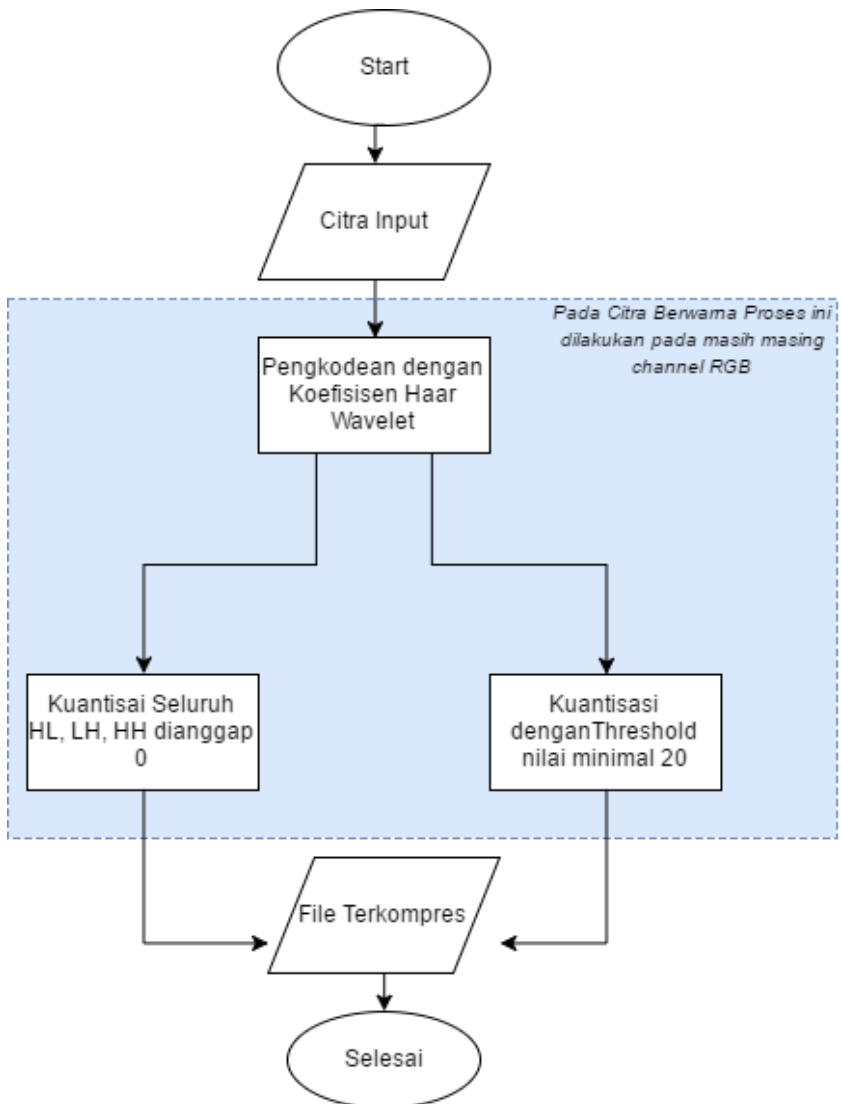
- Baris 1 : $[(10+10)/2 \ (20+20)/2 \ (10-10)/2 \ (20-20)/2] = [10 \ 20 \ 0 \ 0]$
- Baris 2 : $[(10+10)/2 \ (10+10)/2 \ (10-10)/2 \ (10-10)/2] = [10 \ 10 \ 0 \ 0]$

- Baris 3 : $[(50+50)/2 (30+30)/2 (50-50)/2 (30-30)/2] = [50 \ 30 \ 0 \ 0]$
- Baris 4 : $[(50+50)/2 (30+30)/2 (50-50)/2 (30-30)/2] = [50 \ 30 \ 0 \ 0]$

Gambar 3.4 (c) diperoleh dari proses perataan dan pengurangan dari Gambar 3.5 (b):

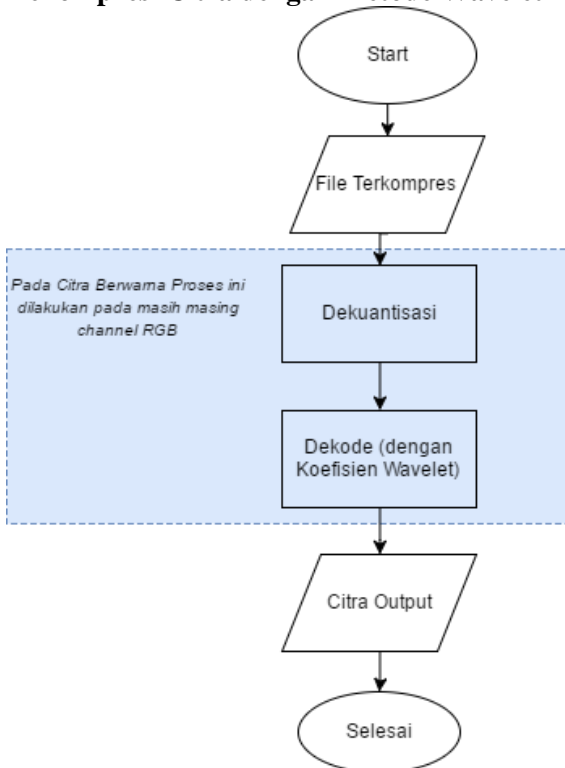
- Kolom 1 : $[(10+10)/2 (50+50)/2 (10-10)/2 (50-50)/2] = [10 \ 50 \ 0 \ 0]$
- Kolom 2 : $[(20+10)/2 (30+30)/2 (20-10)/2 (30-30)/2] = [15 \ 30 \ 5 \ 0]$
- Kolom 3 : $[(0+0)/2 (0+0)/2 (0-0)/2 (0-0)/2] = [0 \ 0 \ 0 \ 0]$
- Kolom 4 : $[(0+0)/2 (0+0)/2 (0-0)/2 (0-0)/2] = [0 \ 0 \ 0 \ 0]$

Seluruh proses dalam satu level dekomposisi akan membagi menjadi dua bentuk sinyal yang disebut *approximation coefficient* dari hasil *lowpass filter* dan *detail coefficient* hasil *highpass filter*. LL atau *lowpass low* disebut juga sebagai *approximation coefficient*, HL atau *Highpass Low* disebut juga sebagai detil horizontal, LH atau *Lowpass Low* disebut juga sebagai detil vertical, HH atau *Highpass High* disebut juga sebagai detil diagonal. Sebagai contoh dari Gambar 3.4 (c) hasil Aproksimasi atau LLnya adalah ($[10 \ 50 \ 0 \ 0]$; $[10 \ 10 \ 0 \ 0]$). Selanjutnya yaitu kuantisasi, hasil transformasi (selain komponen aproksimasi) dilakukan proses kuantisasi. Tujuan dari proses ini adalah mengurangi variasi data pada komponen hasil transformasi selain aproksimasi. Pada Tugas Akhir kali ini terdapat 2 macam proses kuantisasi, yaitu menghilangkan seluruh parameter selain LL (Aproksimasi) atau dengan kata lain koefisien lainnya dianggap bernilai 0, dan proses lnlainnya yaitu menghilangkan atau menganggap 0 nilai pixel apabila nilai pixel lebih kecil dari 20. Proses terakhir yaitu penyimpanan hasil transformasi. Untuk citra berwarna (*color*) proses wavelet mulai dari pengkodean hingga kuantisasi dilakukan 3x yaitu untuk masing masing channel Red Green Blue (RGB) Gambaran proses kompresi wavelet terlihat pada Gambar 3.6.



Gambar 3.6 Flowchart Kompresi Wavelet

3.2.2 Dekompresi Citra dengan Metode Wavelet



Gambar 3.7 Flowchart Dekompresi Wavelet

Algoritma dekomposisi wavelet seperti yang ditunjukkan pada Gambar 3.7 dapat dijelaskan sebagai berikut [1]:

1. Membaca data data atau file terkompres yang disimpan.
2. Langkah pertama dalam proses dekomposisi yaitu melakukan dekuantisasi (kuantisasi balik) untuk mengembalikan data yang sebelumnya di kuantisasi saat proses kompresi. Proses ini akan mengembalikan

- nilai komponen-komponen detail vertikal (HL), horizontal (LH), dan diagonal (HH).
3. Menggabungkan seluruh komponen hasil tahap 2 dengan komponen aproksimasi.
 4. Melakukan transformasi balik (Invers Wavelet) terhadap komponen-komponen hasil tahap 3.
 5. Proses terakhir adalah menyimpan seluruh hasil invers sebagai citra output yang bersifat lossy.

3.2.3 Kompresi Citra dengan Metode Fractal

Fraktal adalah obyek yang memiliki kemiripan dirinya-sendiri (*self-similarity*) namun dalam skala yang berbeda. Ini artinya, bagian-bagian dari obyek akan tampak sama dengan obyek itu sendiri bila dilihat secara keseluruhan. Gambaran kompresi fraktal ditampilkan pada Gambar 3.10.

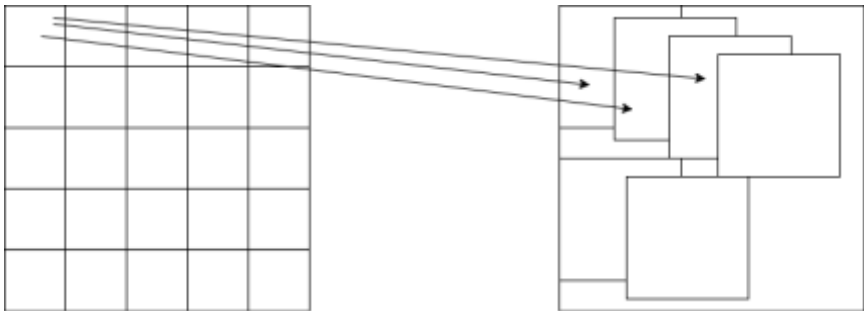
Langkah pertama yang dilakukan dalam proses kompresi adalah membagi citra asli (Gambar 3.8) menjadi dua blok yaitu blok jelajah (*range*) dan blok domain (*ranah*). Selanjutnya dilakukan pembentukan sejumlah blok yang berukuran sama dan tidak saling beririsan dari citra asli, yang disebut blok jelajah (*range*). Untuk menyederhanakan masalah, blok jelajah diambil berbentuk bujursangkar. Pada tugas akhir ini blok *range* yang dibuat berukuran 8x8. Selanjutnya dibuat sejumlah blok berukuran sama yang lebih besar dibanding blok *range* namun dapat beririsan satu sama lain, blok-blok tersebut disebut sebagai blok domain, pada tugas akhir ini ukuran blok domain yang digunakan yaitu 16 x 16.

Langkah selanjutnya yaitu mencari pasangan dari kedua blok ini (Gambar 3.9) dengan menghitung nilai kemiripan untuk setiap blok *range* ke setiap blok domain. Untuk menghitung kemiripan antar dua buah blok matrix digunakan rumus *rms* (*root mean square*). Untuk setiap blok domain sebelum dilakukan perhitungan *rms*, dilakukan penskalaan yaitu menghitung rata-rata nilai pixel untuk setiap 2x2 blok pada

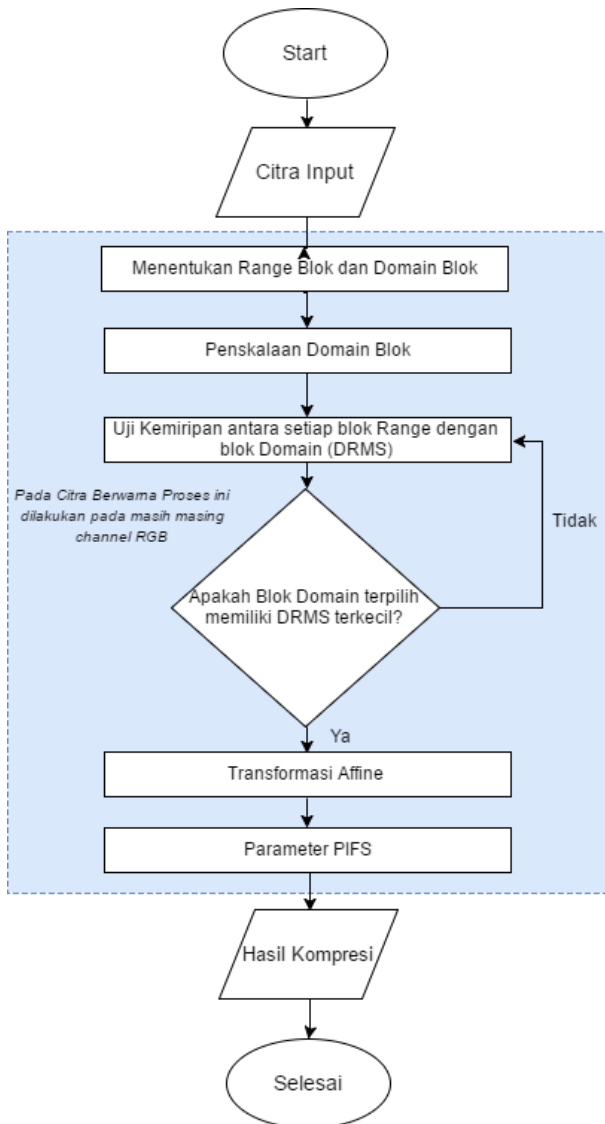
blok domain sebagai nilai 1 pixel sehingga ukuran blok domain hasil penskalaan sama dengan ukuran blok range.

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

Gambar 3.8 Contoh matrix dari citra asli



Gambar 3.9 Ilustrasi pencarian nilai pasangan blok range dan blok domain



Gambar 3.10 Flowchart Kompresi Fraktal

Pada kompresi fraktal proses terlama adalah saat uji kemiripan, karena untuk setiap range blok harus di hitung drmsnya dengan setiap domain blok. Sebagai contoh citra 512x 512 memiliki 4,096 blok range (ukuran 8x8) dan 247,009 blok domain (ukuran 16x16 beririsan), yang artinya ada $4,096 \times 247,009 = 1,011,748,864$ pasangan blok domain dan range domain yang akan dihitung drmsnya, oleh karena itu proses ini memerlukan waktu yang sangat lama. Langkah terakhir yaitu menyimpan koefisien transformasi affine sebagai PIFS untuk setiap blok domain yang menjadi pasangan dari range blok sebagai file terkompres. Parameter yang disimpan yaitu si , oi , ei , fi dimana Parameter si menyatakan faktor kontras *pixel* (seperti tombol kontras di TV). Bila si bernilai 0, maka *pixel* menjadi gelap, bila si sama dengan 1 kontrasnya tidak berubah; antara 0 dan 1 *pixel* berkurang kontrasnya, di atas 1 kontrasnya bertambah. Parameter oi menyatakan offset kecerahan (*brightness*) *pixel* (seperti tombol kecerahan di TV). Nilai oi positif mencerahkan gambar dan nilai oi negatif menjadikannya gelap. Parameter ei dan fi mudah dihitung karena keduanya menyatakan pergeseran sudut kiri blok ranah ke sudut kiri blok jelajah yang bersesuaian.

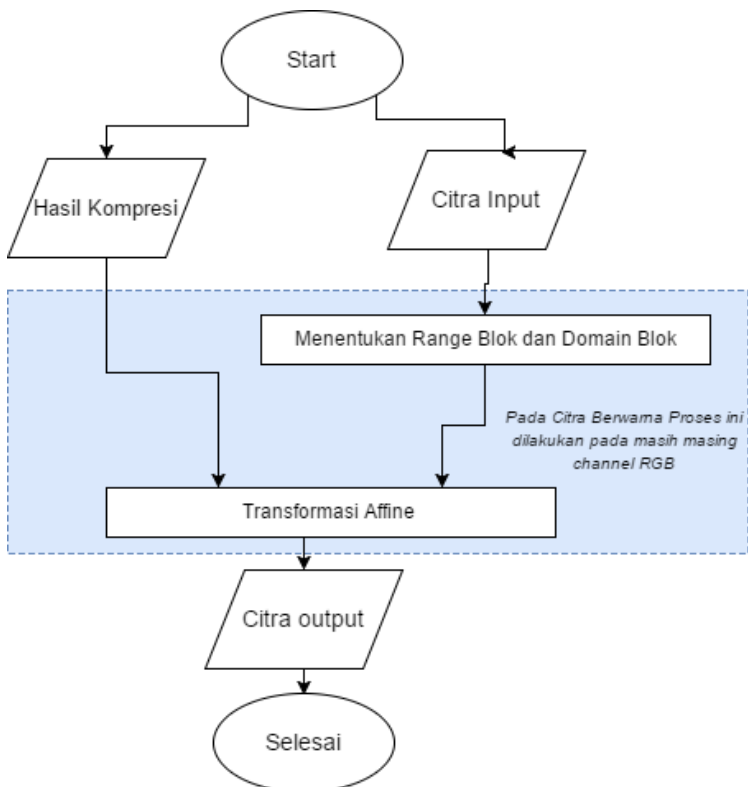
Rekonstruksi (*decoding*) citra dilakukan dengan merekonstruksi dari citra awal sembarang. Selama proses pemulihan, setiap *IFS* lokal mentransformasikan sekumpulan blok ranah menjadi sekumpulan blok jelajah. Karena blok jelajah tidak saling beririsan dan mencakup keseluruhan *pixel* citra, maka gabungan seluruh blok jelajah menghasilkan citra titik tetap yang menyerupai citra semula.

3.2.4 Dekompresi Citra dengan Metode Fractal

Proses rekonstruksi fraktal ditunjukkan pada Gambar 3.11. Adapun algoritma untuk dekompresi kompresi citra digital dengan fraktal adalah [5] :

1. Membaca parameter PIFS yang tersimpan sebagai file terkompres.

2. Membaca citra sembarang (Citra lainnya) yang memiliki ukuran sama dengan citra inputnya.
3. Pada file citra lainnya dilakukan proses pembentukan Range blok berukuran 8x8 tak beraturan.
4. Langkah selanjutnya yaitu pembentukan ulang citra dengan menggunakan transformasi affine pada seluruh Range blok.
5. Hasil transformasi merupakan citra output program yang bersifat lossy.



Gambar 3.11 *Flowchart* Dekompresi Fraktal

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi berupa kode sumber untuk membangun program. Sebelum masuk ke penjelasan implementasi, akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Implementasi

Implementasi Studi Kerja Metode *Lossy Compression* Pada Citra Medis Menggunakan Algoritma Kompresi Berbasis Fraktal dan Wavelet menggunakan spesifikasi perangkat keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Spesifikasi lingkungan implementasi

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel(R) Core(TM) i3-3240 CPU @ 3.40GHz 3.40 GHz
	Memori	8 GB 1600
Perangkat Lunak	Sistem Operasi	Windows 8.1 Pro
	Perangkat Pengembang	MATLAB R2014a

4.2 Implementasi

Pada sub bab implementasi akan menjelaskan bagaimana pembangunan perangkat lunak secara detail dan menampilkan hasil kompresi dengan Wavelet dan Fractal. Seperti yang telah dijelaskan pada bab perancangan sistem, bahwa hal yang dilakukan dalam Implementasi Studi Kerja Metode *Lossy*

Compression Pada Citra Medis Menggunakan Algoritma Kompresi Berbasis Fraktal dan Wavelet adalah melakukan pengkompresian citra dengan metode Fractal dan Wavelet.

4.2.1 Kompresi dan Dekompresi Citra dengan Fractal

Kompresi Fractal akan mengkompresi *file* berformat *.bmp. Proses dimulai dengan menentukan jenis *image* masukan apakah berupa *image* grayscale atau citra warna (RGB). Jika *image* inputan diketahui berupa *image* grayscale maka program akan langsung membagi *pixel-pixel* pada *image* tersebut menjadi dua blok, yaitu blok *domain* dan blok *range*. Dalam hal ini blok *range* dibuat berukuran 8 x 8 dan karena blok *domain* harus dua kali blok *range* maka blok *domain* dibuat berukuran 16 x 16. Pembagian *image* masukan kedalam blok – blok domain dan range dapat dilihat pada kode sumber 4.1 dan 4.2 dengan menggunakan bahasa pemrograman Matlab.

```

1 image = imread('CitraMedisGray1.bmp');
2 IG_d = double(image);
3 % Membuat blok 8x8 pada Range block
4 szM = size(IG_d);
5 nb = szM ./ bs; % Banyaknya block tiap dimensi
6 Range=mat2cell(IG_d, repmat(bs(1),1,nb(1)),
7 repmat(bs(2),1,nb(2)));
8 C2 = cellfun(@(x) min(x(:)), Range, 'un', 0);
9 M2 = cell2mat(C2);

```

Kode Sumber 4.1 Proses pembagian Blok Range


```

1  % Membuat 16x16 blok beririsan untuk Domain
2  block
3  for i=1:bar-15
4      for j=1:kol-15
5          thresholded = zeros(16,16);
6          for k=1:16
7              for l=1:16
8                  thresholded(k,l)=IG_d(i+k-1,
9                      j+l-1);
10             end
11         end
12         IG_outd{i,j}= thresholded;
13     end
14 end

```

Kode Sumber 4.2 Proses pembagian Blok Domain

Sebelum proses pencarian kemiripan dimulai, setiap blok domain diskalakan sehingga ukurannya sama dengan ukuran blok range. Penskalaan ini dimaksudkan agar jarak antara blok range dan blok domain mudah dihitung. Penskalaan dilakukan dengan menjadikan 2x2 buah *pixel* menjadi satu buah *pixel*. Nilai satu buah *pixel* tersebut adalah rata-rata nilai keempat *pixel* seperti yang terlihat pada kode sumber 4.3

```

1  function [IG_out] = downsampling(image)
2  %menggunakan treshold 2x2
3  IG_d = double(image);
4  [bar,kol] = size(IG_d);
5  IG_out = double(zeros(bar/2, kol/2));
6  for i=1:2:bar-1
7      for j=1:2:kol-1
8          thresholded = zeros(2,2);
9          for k=1:2
10             for l=1:2
11                 thresholded(k,l)=IG_d(i+k-1,
12                     j+l-1);
13             end
14         end

```

15	IG_out((i+1)/2,j+1)/2)=
16	mean2(thresholded);
17	end
18	end

Kode Sumber 4.3 Penskalaan Blok Domain

Dengan menggunakan RMS kemudian dicari kemiripan antara blok range dan blok domain yang sudah diskalakan menjadi $\frac{1}{2}$ ukuran blok domain semula, setelah diperoleh drms dari masing – masing blok domain terhadap blok range, kemudian dipilih nilai drms yang terkecil yang menandakan persentase kemiripan yang terbesar. Pencarian kemiripan antara blok range dan blok domain dapat dilihat pada Kode Sumber 4.4.

1	<code>function [drms] = rms(range, domain)</code>
2	<code>%Menghitung rms untuk antara range dan domain</code>
3	<code>[size1, size2] = size(domain);</code>
4	<code>numberOfPixels = size1*size2;</code>
5	<code>temp_mutlak = 0;</code>
6	<code>temp_mutlak2 = 0;</code>
7	<code>mutlak = 0;</code>
8	<code>rmsmax = 0;</code>
9	<code>for i=1:size1</code>
10	<code>for j=1:size2</code>
11	<code>temp_mutlak = range(i,j)-domain(i,j);</code>
12	<code>temp_mutlak2 = temp_mutlak^2;</code>
13	<code>mutlak = mutlak + temp_mutlak2;</code>
14	<code>end</code>
15	<code>end</code>
16	<code>drms = sqrt((1/numberOfPixels) * mutlak);</code>
17	<code>end</code>

Kode Sumber 4.4 Pencarian Kemiripan antara Blok Range dan Blok Domain

Blok yang terpilih akan diproses menggunakan Transformas Affine antara blok range dan blok domain yang terpilih untuk memperoleh nilai yang akan disimpan dalam

koefisien PIFS (s_i , o_i , e_i dan f_i) seperti terlihat pada Kode Sumber 4.5.

```

1  %Menyimpan parameter PIFS
2  for i=1:bar_range
3      for j=1:kol_range
4          [s,o]=brighcont(Range{i,j},
5              domain_poll{i,j});
6              Index = index_poll{i,j};
7
8          e = Index(1,1);
9          f = Index(1,2);
10
11          affine_param{i,j} = [e f s o];
12      end
13  end

```

Kode Sumber 4.5 Transformasi Affine blok range dan blok domain

Selanjutnya untuk proses dekompresi dilakukan dengan cara membentuk kembali citra dengan menggunakan parameter PIFS yang ada, parameter tersebut dimasukan ke dalam persamaan transformasi affine untuk tiap tiap reng blok untuk mengembalikan nilai pixel citra semula. Proses dekompresi dapat di lihat pada Kode Sumber 4.6.

```

1  for i=1:bar
2      for j=1:kol
3          N=(C{i,j});
4          param=cell2mat(N(:));
5          M=double(domain_new{i,j});
6          e=param(1,1);
7          f=param(2,1);
8          s=param(3,1);
9          o=param(4,1);
10         T=[0.5, 0.5, 0;0, 0.5, 0;0, 0, s];
11         for k=1:8
12             for l=1:8

```

```

13         xyz=[k;1;M(k,1)];
14         efo=[e;f;o];
15         W=(T*xyz)+efo;
16         pixel=W(3,1);
17         temp(k,1)=pixel;
18     end
19 end
20 tampung{i,j} = temp;
21 end
22 end

```

Kode Sumber 4.6 Proses Dekompresi Algoritma Fraktal

4.2.2 Kompresi dan Dekompresi Citra dengan Wavelet

Proses dimulai dengan menentukan jenis *image* masukan apakah berupa *image* grayscale atau citra warna (RGB). Jika *image* inputan diketahui berupa *image color* maka program akan melakukan penghitungan pada 3 layer, yaitu layer *Red*, layer *Green*, dan Layer *Blue*. Pada masing masing layer RGB akan dicari koefisien Aproximate atau LL, Detail Vertikal atau HL, Detail Horizontal atau LH, dan Detail Diagonal atau HH. Pencarian koefisien A, DV, DH, dan DD ditunjukkan pada kode sumber 4.7 dengan menggunakan bahasa pemrograman Matlab.

```

1 %Menhitung koefisien A,DH DV,DD untuk setiap
2 chanel RGB
3 [xar, xhr, xvr, xdr] = dwt2(X(:,:,1),'haar');
4 [xag, xhg, xvg, xdg] = dwt2(X(:,:,2),'haar');
5 [xab, xhb, xvb, xdb] = dwt2(X(:,:,3),'haar');
6 xa(:,:,1)=xar; xa(:,:,2)=xag; xa(:,:,3)= xab;
7 xh(:,:,1)=xhr; xh(:,:,2)=xhg; xh(:,:,3)= xhb;
8 xv(:,:,1)=xvr; xv(:,:,2)=xvg; xv(:,:,3)= xvb;
9 xd(:,:,1)=xdr; xd(:,:,2)=xdg; xd(:,:,3)= xdb;

```

Kode Sumber 4.7 Menghitung nilai A,DH DV,DD untuk setiap chanel RGB

Proses selanjutnya yaitu kuantisasi, kuantisasi merupakan proses pengurangan komponen komponen selain aproksimasi. Kode sumber 4.8 menunjukkan kuantisasi dan

pembentukan matrix baru sebagai pengganti nilai detail horizontal, vertikal, dan diagonal yang keseluruhannya dianggap 0.

1	%kuantisasi
2	LH = zeros(size(xh));
3	HL = zeros(size(xv));
4	HH = zeros(size(xd));

Kode Sumber 4.8 Kuantisasi nilai xh, xv, xd.

Kode sumber 4.9 menunjukkan kuantisasi dan pembentukan matrix baru sebagai pengganti nilai detail horizontal, vertikal, dan diagonal yang memiliki nilai lebih kecil dari 20 dianggap 0.

1	function [kuantisasi] = quantize(detail)
2	[size1,size2] = size(detail);
3	threshold = 0;
4	temp = 0;
5	k=1;
6	for i=1:size1
7	for j=1:size2
8	threshold = detail(i,j);
9	if threshold >=2
10	temp(k,1)= threshold;
11	temp(k,2) = i;
12	temp(k,3) = j;
13	k= k+1;
14	end
15	end
16	end
17	kuantisasi = temp;
18	end

Kode Sumber 4.9 Kuantisasi dengan nilai treshhold minimal 20.

Tahap terakhir yaitu penyimpanan parameter Aproksimasi serta index dan nilai pixel threshold sebagai file terkompres.

Untuk proses dekompresi citra dengan algoritma wavelet seperti pada Kode sumber 4.10 yaitu dimulai dari proses dekuantisasi atau kuantisasi balik, dengan mengembalikan parameter detail (LH, HL dan HH).

```

1  Function [dekuantisasi]=dequantize(detail,aprox)
2      [size1,size2] = size(aprox);
3      [bar,kol] = size(detail);
4      temp = zeros(size1,size2);
5      for i=1:bar
6          x=detail(i,2);
7          y=detail(i,3);
8          temp(x,y) = detail(i,1);
9      end
10     dekuantisasi = temp;
11 end

```

Kode Sumber 4.10 Proses Dekuantisasi.

Tahap terakhir yaitu pembentukan kembali citra dengan invers wavelet. Yaitu menggunakan koefisien aproksimasi beserta detailnya (LH, HL, dan HH) ditunjukkan pada Kode sumber 4.11.

```

1  [THR, SORH,KEEPAPP]= ddencmp('cmp','wv',X);
2
3  komp_horizontal = wdencomp('gb1', xLH,wname,
4  1, THR, SORH, KEEPAPP);
5  komp_vertical = wdencomp('gb1', xHL,wname, 1,
6  THR, SORH, KEEPAPP);
7  komp_diagonal = wdencomp('gb1', xHH,wname, 1,
8  THR, SORH, KEEPAPP);
9  rekonstruksi_citra = idwt2(Aproximasi,
10 komp_horizontal, komp_vertical,
11 komp_diagonal, wname, size(X));
12 citra terkompres = uint8(rekonstruksi citra);

```

Kode Sumber 4.11 Pembentukan citra dari file terkompres.

BAB V

UJI COBA DAN EVALUASI

Pada Bab ini akan dijelaskan mengenai skenario uji coba pada perangkat lunak yang telah dibangun. Setelah itu, hasil uji coba akan dianalisa kinerjanya sehingga dapat diketahui kriteria, kelebihan dan kekurangan dari Kompresi citra dengan Fractal dan Wavelet. Secara garis besar, bab ini berisikan pembahasan mengenai lingkungan pengujian, data pengujian, dan uji kinerja.

5.1 Lingkungan Pengujian

Lingkungan pengujian pada Studi Kerja Metode *Lossy Compression* Pada Citra Medis Menggunakan Algoritma Kompresi Berbasis Fraktal dan Wavelet menggunakan spesifikasi perangkat keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Lingkungan Uji Coba

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Prosesor: Intel® Core™ i3-3240 CPU @ 3.40 GHz
	Memori	8 GB 1600
Perangkat Lunak	Sistem Operasi	Windows 8.1 Pro
	Perangkat Pengembang	MATLAB R2014a

5.2 Data Pengujian

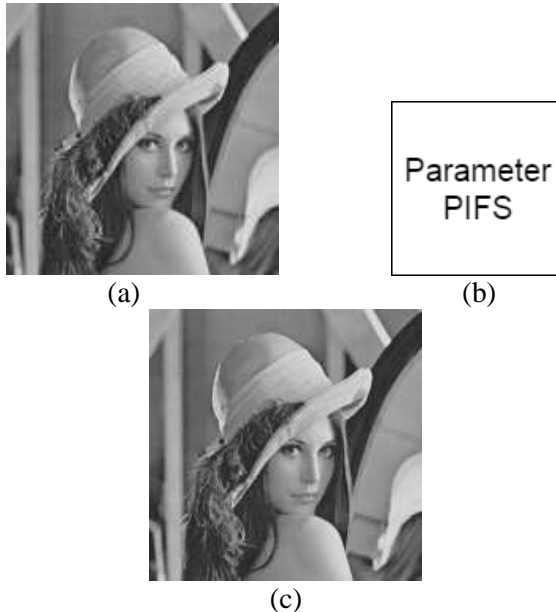
Seperti yang telah dijelaskan sebelumnya, data yang digunakan pada Tugas Akhir ini adalah data citra Medis. Pada Tugas Akhir ini, data awalan adalah data *image* yang diambil dari data storage. Sedangkan untuk skenario uji coba, dataset akan dibagi menjadi 2 data yang berbeda yaitu:

1. Dataset 1 merupakan data *image* grayscale yang terdiri dari 5 *image*.
2. Dataset 2 merupakan data *image* Warna (RGB) yang terdiri dari 5 *image*.

Data awalan ini kemudian akan masuk ke proses Kompresi citra dengan Fractal dan Wavelet. Selanjutnya citra hasil kompresi akan dibandingkan dengan citra aslinya, sehingga dapat diperoleh perbandingan hasil citra asli dengan citra hasil kompresi Fractal dan Wavelet



Gambar 5.1 Gambaran hasil proses kompresi Wavelet



Gambar 5.2 Gambaran hasil proses kompresi Fraktal

5.3 Skenario Uji Coba

Sebelum melakukan uji coba, perlu ditentukan skenario yang akan digunakan dalam uji coba. Melalui skenario uji coba ini, akan dianalisa hasil kompresi setiap *image* terhadap metode Kompresi Fraktal dan Wavelet, bagaimana kualitas hasil kompresi pada masing-masing metode Kompresi disetiap skenario dan perbandingan performa antara metode Kompresi Fraktal dan Wavelet yang memiliki hasil paling baik. Pada percobaan kali ini akan digunakan beberapa level wavelet yaitu Level 1, Level 2, dan Level 3. Pada Tugas Akhir ini, terdapat berbagai macam skenario uji coba, yaitu:

1. Kompresi citra grayscale dengan 3 data masukan citra medis.
2. Kompresi citra grayscale dengan data masukan citra Lena.
3. Kompresi citra grayscale dengan data masukan citra Baboon.
4. Kompresi citra warna (RGB) dengan 3 data masukan citra medis.
5. Kompresi citra warna (RGB) dengan data masukan citra Lena.
6. Kompresi citra warna (RGB) dengan data masukan citra Baboon.
7. Kompresi dan PSNR dengan format PNG, JPG, dan GIF.

5.3.1 Skenario Uji Coba Kompresi citra grayscale dengan data masukan citra medis.

Pada skenario uji coba yang pertama, data masukan yang digunakan adalah citra medis grayscale berukuran 512 x 512 pixel. Citra medis input selanjutnya diproses dengan menggunakan metode kompresi wavelet dan fraktal. Hasil uji coba kompresi dengan metode wavelet dan fraktal dapat dilihat pada tabel 5.2 dan 5.3.

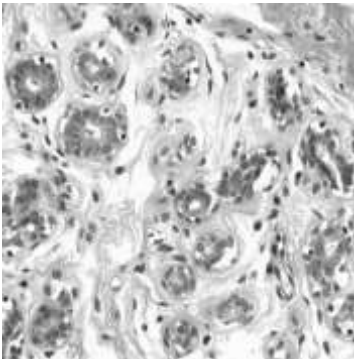
Tabel 5.2 Hasil kompresi citra medis dengan Wavelet

Jenis Kompresi	Wavelet LV1	Wavelet LV2	Wavelet LV3	Wavelet LV1 Thres	Wavelet LV2 Thres	Wavelet LV3 Thres
Nama File (.bmp)	GrayMedisDataset 1	GrayMedisDataset 1	GrayMedisDataset 1	GrayMedisDataset 1	GrayMedisDataset 1	GrayMedisDataset 1
Size Awal (KB)	263.22	263.22	263.22	263.22	263.22	263.22
Time(s)	0.261	0.348	0.116	0.292	0.090	0.30
MSE	85.28	299.4	672.74	56.93	198.84	362.88
Size Akhir(KB)	149.82	42.42	12.01	212.45	106.16	95.32
PSNR	28.82	23.36	19.85	30.57	25.14	22.53
CR	1.75	6.2	21.9	1.23	2.48	2.76
BPP (*10 ⁴)	5.71	1.61	45.85	8.16	4.05	3.63

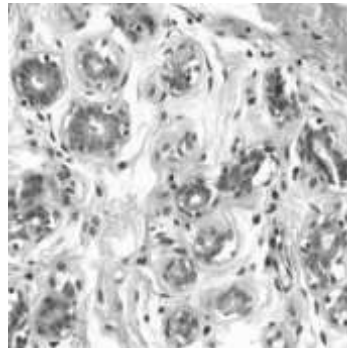
Tabel 5.3 Hasil kompresi citra medis dengan Fraktal

No	Nama File (.bmp)	Size Awal (KB)	Fraktal					
			Time (s)	MSE	Size Akhir (KB)	PSNR	CR	BPP (*10 ⁻⁴)
1	GrayMedis Dataset1	258	6836.1	132.10	105	21.14	2.45	4.06
2	GrayMedis Dataset2	258	6906.2	133.11	105	21.14	2.45	4.06
3	GrayMedis Dataset3	258	6930.9	128.7	104	21.28	2.48	4.08

Gambar berikut merupakan citra hasil kompresi dengan metode fraktal dan wavelet. Hasil kompresi citra wavelet dan fraktal dapat dilihat pada Gambar 5.3 dan Gambar 5.4.

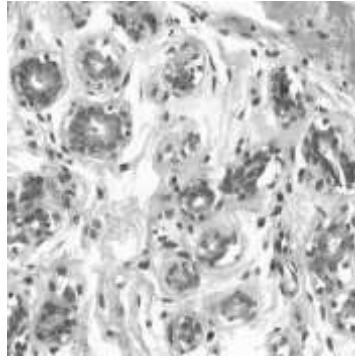
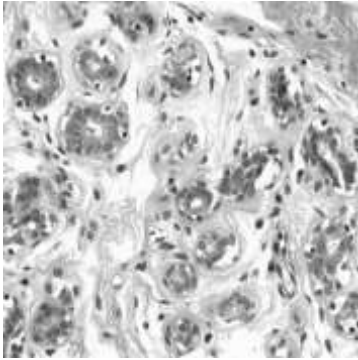


a. Citra asli



b. Hasil kompresi

Gambar 5.3 Hasil Kompresi Citra Grayscale Medis dengan Wavelet



a. Citra asli

b. Hasil kompresi

Gambar 5.4 Hasil Kompresi Citra Grayscale Medis dengan Fraktal

5.3.2 Skenario Uji Coba Kompresi citra grayscale dengan data masukan citra Lena.

Pada skenario uji coba yang kedua, data masukan yang digunakan adalah citra Lena grayscale berukuran 512 x 512 pixel. Citra input selanjutnya diproses dengan menggunakan metode kompresi wavelet dan fraktal. Hasil uji coba kompresi dengan metode wavelet dan fraktal dapat dilihat pada tabel 5.4 dan 5.5.

Tabel 5.4 Hasil kompresi citra Lena dengan Wavelet

Jenis Kompresi	Wavelet LV1	Wavelet LV2	Wavelet LV3	Wavelet LV1 Thres	Wavelet LV2 Thres	Wavelet LV3 Thres
Nama File (.bmp)	GrayLena	GrayLena	GrayLena	GrayLena	GrayLena	GrayLena
Size Awal (KB)	263.22	263.22	263.22	263.22	263.22	263.22
Time(s)	0.2615	0.1756	0.1333	0.279	0.170	0.058
MSE	45.51	132.17	279.72	30.16	74.40	144.05

Size Akhir(KB)	140.05	40.97	11.63	152.76	68.87	49.85
PSNR	31.54	26.91	24.66	33.33	29.41	26.54
CR	1.87	6.42	22.61	1.72	3.82	5.28
BPP (*10 ⁴)	5.43	1.56	44.39	5.82	2.62	1.90

Tabel 5.5 Hasil kompresi citra Lena dengan Fraktal

	Nama File (.bmp)	Size Awal (KB)	Fraktal					
			Time (s)	MSE	Size Akhir (KB)	PSNR	CR	BPP (*10 ⁻⁴)
1	GrayLena	258	6924.9	122.51	105	22.62	2.45	4.01

Gambar berikut merupakan citra hasil kompresi dengan metode fraktal dan wavelet. Hasil kompresi citra wavelet dan fraktal dapat dilihat pada Gambar 5.5 dan Gambar 5.6.



a. Citra asli



b. Hasil kompresi

Gambar 5.5 Hasil Kompresi Citra Grayscale Lena dengan Wavelet



a. Citra asli



b. Hasil kompresi

Gambar 5.6 Hasil Kompresi Citra Grayscale Lena dengan Fraktal

5.3.3 Skenario Uji Coba Kompresi citra grayscale dengan data masukan citra Baboon.

Pada skenario uji coba yang ketiga, data masukan yang digunakan adalah citra Baboon grayscale berukuran 256 x 256 pixel. Citra input selanjutnya diproses dengan menggunakan metode kompresi wavelet dan fraktal. Hasil uji coba kompresi dengan metode wavelet dan fraktal dapat dilihat pada tabel 5.6 dan 5.7.

Tabel 5.6 Hasil kompresi citra Baboon dengan Wavelet

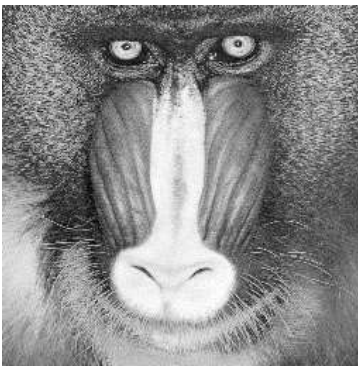
Jenis Kompresi	Wavelet LV1	Wavelet LV2	Wavelet LV3	Wavelet LV1 Thres	Wavelet LV2 Thres	Wavelet LV3 Thres
Nama File (.bmp)	Gray Babon	Gray Babon	Gray Babon	Gray Babon	Gray Babon	Gray Babon
Size Awal (KB)	66.09	66.09	66.09	66.09	66.09	66.09
Time(s)	1.325	0.024	0.039	0.10	0.11	0.04
MSE	97.5	147.96	190.70	67.96	102.12	122.14

Size Akhir(KB)	35.43	10.31	3.00	44.38	24.30	19.91
PSNR	28.24	26.42	25.32	29.80	28.04	27.26
CR	1.86	6.40	22.00	1.49	2.71	3.31
BPP (*10 ⁴)	5.40	1.57	0.45	6.77	3.70	3.03

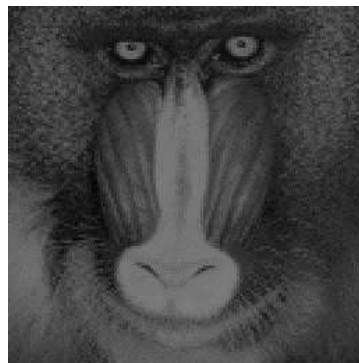
Tabel 5.7 Hasil kompresi citra Baboon dengan Fraktal

	Nama File (.bmp)	Size Awal (KB)	Fraktal					
			Time (s)	MSE	Size Akhir (KB)	PSNR	CR	BPP (*10 ⁻⁴)
1	Gray Baboon	65	421.34	131.10	29	21.17	2.24	4.49

Gambar berikut merupakan citra hasil kompresi dengan metode fraktal dan wavelet. Hasil kompresi citra wavelet dan fraktal dapat dilihat pada Gambar 5.7 dan Gambar 5.8.

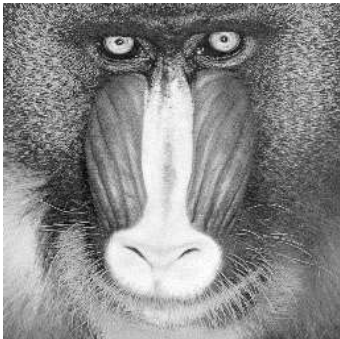


a. Citra asli

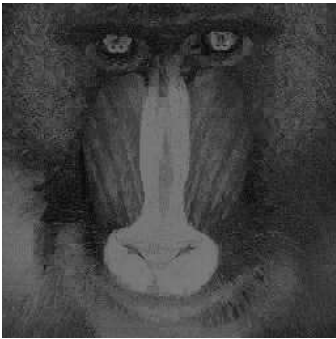


b. Hasil Kompresi

Gambar 5.7 Hasil Kompresi Citra Greyscale Baboon dengan Wavelet



a. Citra asli



b. Hasil Kompresi

Gambar 5.8 Hasil Kompresi Citra Greyscale Baboon dengan Fraktal

5.3.4 Skenario Uji Coba Kompresi citra color dengan data masukan citra medis.

Pada skenario uji coba yang keempat, data masukan yang digunakan adalah citra medis color berukuran 512 x 512. Citra medis input selanjutnya diproses dengan menggunakan metode kompresi wavelet dan fraktal. Hasil uji coba kompresi dengan metode wavelet dan fraktal dapat dilihat pada tabel 5.8 dan 5.9.

Tabel 5.8 Hasil kompresi citra color medis dengan Wavelet

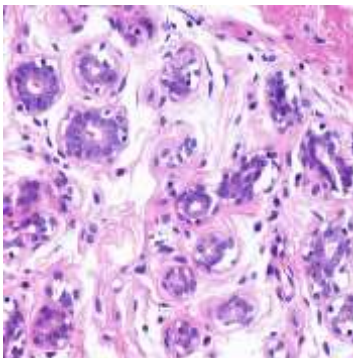
Jenis Kompresi	Wavele t LV1	Wavele t LV2	Wavele t LV3	Wavele t LV1 Thres	Wavele t LV2 Thres	Wavele t LV3 Thres
Nama File (.bmp)	ColorM edisDat aset1	ColorM edisDat aset1	ColorM edisDat aset1	ColorM edisDat aset1	ColorM edisDat aset1	ColorM edisDat aset1
Size Awal (KB)	786.48	786.48	786.48	786.48	786.48	786.48
Time(s)	0.722	0.369	0.283	0.669	0.712	0.809
MSE	25.18	87.54	194.79	16.55	49.70	104.61
Size Akhir(KB)	434.37	124.19	35.06	499.51	283.36	247.01
PSNR	29.34	23.93	20.46	31.16	26.39	23.16

CR	1.81	6.33	22.43	1.57	2.77	3.18
BPP (*10 ⁴)	5.52	1.57	44.58	6.35	3.60	3.14

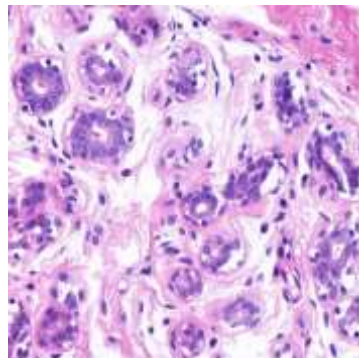
Tabel 5.9 Hasil kompresi citra color medis dengan Fraktal

No	Nama File (.bmp)	Size Awal (KB)	Fraktal					
			Time (s)	MSE	Size Akhir (KB)	PSNR	CR	BPP (*10 ⁴ -4)
1	ColorMedisData t1	769	27119.98	128.43	315	22.01	2.52	4.01
2	ColorMedisData t2	769	22160.87	134.09	313	21.19	2.54	3.98
3	ColorMedisData t3	769	20637.79	131.21	312	21.09	2.55	3.96

Gambar berikut merupakan citra hasil kompresi dengan metode fraktal dan wavelet. Hasil kompresi citra wavelet dan fraktal dapat dilihat pada Gambar 5.9 dan Gambar 5.10.

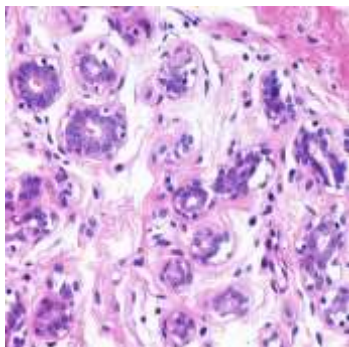


a. Citra asli

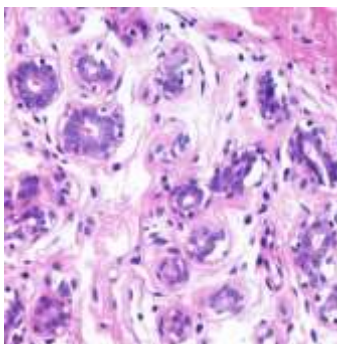


b. Hasil kompresi

Gambar 5.9 Hasil Kompresi Citra Color Medis dengan Wavelet



a. Citra asli



b. Hasil kompresi

Gambar 5.10 Hasil Kompresi Citra Color Medis dengan Fraktal

5.3.5 Skenario Uji Coba Kompresi citra color dengan data masukan citra Lena.

Pada skenario uji coba yang kelima, data masukan yang digunakan adalah citra Lena color berukuran 512 x 512 pixel. Citra input selanjutnya diproses dengan menggunakan metode kompresi wavelet dan fraktal. Hasil uji coba kompresi dengan metode wavelet dan fraktal dapat dilihat pada tabel 5.10 dan 5.11.

Tabel 5.10 Hasil kompresi citra Lena dengan Wavelet

Jenis Kompresi	Wavelet LV1	Wavelet LV2	Wavelet LV3	Wavelet LV1 Thres	Wavelet LV2 Thres	Wavelet LV3 Thres
Nama File (.bmp)	ColorLena	ColorLena	ColorLena	ColorLena	ColorLena	ColorLena
Size Awal (KB)	196.66	196.66	196.66	196.66	196.66	196.66
Time(s)	0.27	0.102	0.096	2.37	0.24	0.22
MSE	25.3	70.33	147.30	14.60	37.6	144.05
Size Akhir(KB)	106.48	31.18	8.87	124.74	66.16	54.34
PSNR	29.22	24.80	21.76	31.55	27.6	24.75
CR	1.84	6.30	22.15	1.57	2.97	3.61
BPP (*10 ⁴)	5.41	1.58	0.45	6.34	3.36	2.76

Tabel 5.11 Hasil kompresi citra Lena dengan Fraktal

	Nama File (.bmp)	Size Awal (KB)	Fraktal					
			Time (s)	MSE	Size Akhir (KB)	PSNR	CR	BPP (*10 ⁻⁴)
1	ColorLena	193	2411.54	125,91	315	22,12	0.61	4,44

Gambar berikut merupakan citra hasil kompresi dengan metode fraktal dan wavelet. Hasil kompresi citra wavelet dan fraktal dapat dilihat pada Gambar 5.11 dan Gambar 5.12.



a. Citra asli



b. Hasil kompresi

Gambar 5.11 Hasil Kompresi Citra Color Lena dengan Metode Wavelet

a. Citra asli



b. Hasil kompresi

Gambar 5.12 Hasil Kompresi Citra Color Lena dengan Metode Fraktal

5.3.6 Skenario Uji Coba Kompresi citra color dengan data masukan citra Baboon.

Pada skenario uji coba yang keenam, data masukan yang digunakan adalah citra Baboon color berukuran 256 x 256 pixel. Citra input selanjutnya diproses dengan menggunakan metode kompresi wavelet dan fraktal. Hasil uji coba kompresi dengan metode wavelet dan fraktal dapat dilihat pada tabel 5.12 dan 5.13.

Tabel 5.12 Hasil kompresi citra Baboon dengan Wavelet

Jenis Kompresi	Wavelet LV1	Wavelet LV2	Wavelet LV3	Wavelet LV1 Thres	Wavelet LV2 Thres	Wavelet LV3 Thres
Nama File (.bmp)	Color Baboon	Color Baboon	Color Baboon	Color Baboon	Color Baboon	Color Baboon
Size Awal (KB)	77.27	77.27	77.27	77.27	77.27	77.27
Time(s)	0.220	0.093	0.083	0.97	0.63	4.79
MSE	104.60	185.73	251.32	55.82	97.85	131.08
Size Akhir(KB)	460.93	128.991	35.36	692.10	491.94	452.93
PSNR	23.20	20.66	19.36	25.89	23.45	22.18
CR	0.16	0.59	2.18	0.11	0.15	0.17
BPP (*10 ⁴)	5.86	1.64	0.45	8.8	6.25	5.75

Tabel 5.13 Hasil kompresi citra Baboon dengan Fraktal

	Nama File (.bmp)	Size Awal (KB)	Fraktal					
			Time (s)	MSE	Size Akhir (KB)	PSNR	CR	BPP (*10 ⁻⁴)
1	ColorBaboon	76	1325.75	123,60	315	22,20	0,24	4.42

Gambar berikut merupakan citra hasil kompresi dengan metode fraktal dan wavelet. Hasil kompresi citra wavelet dan fraktal dapat dilihat pada Gambar 5.13 dan Gambar 5.14.



a. Citra asli



b. Hasil kompresi

Gambar 5.13 Hasil Kompresi Citra Color Baboon dengan Wavelet



a. Citra asli



b. Hasil kompresi

Gambar 5.14 Hasil Kompresi Citra Color Baboon dengan Fraktal

5.3.7 Skenario Uji Coba Kompresi dan PSNR dengan format PNG, JPG, dan GIF.

Pada skenario uji coba yang ketujuh, data masukan yang digunakan adalah seluruh dataset. Citra input selanjutnya diproses dengan menyimpan kedalam bentuk PNG, JPG, dan GIF. Hasil uji coba dapat dilihat pada tabel 5.14.

Tabel 5.14 Hasil penyimpanan citra dengan extensi PNG, JPG , dan GIF

Gambar	Size awal (KB)	PNG		JPG		GIF	
		Size Akhir (KB)	PSNR	Size Akhir (KB)	PSNR	Size Akhir (KB)	PSNR
GrayMedisDa taset1	258	163	inf	50	37.76	278	inf
GrayMedisDa taset2	258	157	inf	46	37.01	245	inf
GrayMedisDa taset3	258	159	inf	46	36.47	245	inf
GrayLena	258	148	inf	32	37.8	259	inf
GrayBaboon	65	46	inf	13	31.96	66	inf
ColorMedisDa taset1	769	469	inf	59	31.43	243	inf
ColorMedisDa taset2	769	459	inf	54	30.70	205	inf
ColorMedisDa taset3	769	466	inf	53	30.90	207	inf
ColorLena	193	113	inf	12	32.96	54	inf
ColorBaboon	76	588	inf	76	49.54	222	inf

5.4 Analisa Hasil Uji Coba

Secara umum, berkas citra hasil kompresi mempunyai ukuran yang lebih kecil dari pada citra semula kecuali pada beberapa gambar, Tabel 5.15 menunjukkan perbandingan ukuran

citra sebelum dan sesudah dimampatkan. Dari tabel 5.15 terlihat bahwa Kompresi fraktal menghasilkan kompresi yang kecil dan ukurannya relatif stabil karena ukuran kompresi fraktal bergantung pada besarnya resolusi gambar, namun waktu yang dibutuhkan untuk komputasi kompresi Fraktal jauh lebih lama dibandingkan dengan kompresi wavelet. Hal ini terjadi dikarenakan kompresi fraktal membutuhkan waktu yang lama untuk mencocokkan antara tiap tiap range blok dan domain blok. Pada gambar dengan format penyimpanan PNG dan GIF tidak dapat dilakukan perhitungan PSNR dikarenakan pada kedua file format tersebut tidak ada nilai pixel yang diubah sehingga PSNRnya bernilai *infinite*, sedangkan pada file dengan format penyimpanan JPG dapat dilakukan perhitungan karena file format tersebut melakukan kompresi terhadap gambar yang disimpan dengan algoritma yang ada di dalamnya. Untuk keseluruhan nilai PSNR dengan format penyimpanan file JPG memiliki nilai yang lebih tinggi dibanding algoritma kompresi fraktal dan wavelet dikarenakan ekstensi JPG memiliki beberapa algoritma kompresi sehingga menjadikan nilai PSNRnya lebih baik.

Selanjutnya pada kompresi Wavelet, Level dekomposisi dan nilai threshold pada proses kuantisasi juga sangat mempengaruhi hasil dari kompresi citra. Semakin tinggi level dekomposisinya maka semakin kecil ukuran filenya, namun nilai PSNRnya juga semakin kecil seperti pada tabel 5.15. Untuk threshold, dapat dilihat seperti pada tabel 5.16, semakin besar thresholdnya maka kompresinya akan semakin tinggi namun nilai PSNRnya semakin berkurang (pada kasus ini level dekomposisi wavelet yang digunakan yaitu LV1).

Pada kualitas citra hasil kompresi wavelet dan fraktal (dilihat dari PSNR), kompresi Wavelet memiliki hasil yang lebih baik dibanding fraktal. Untuk citra grayscale ataupun color, kompresi wavelet lebih baik digunakan dibanding fraktal. Perbandingan persentase kompresi dapat dilihat pada tabel 5.17 dan Gambar 5.15.

Tabel 5.15 Perbandingan Ukuran Kompresi Citra

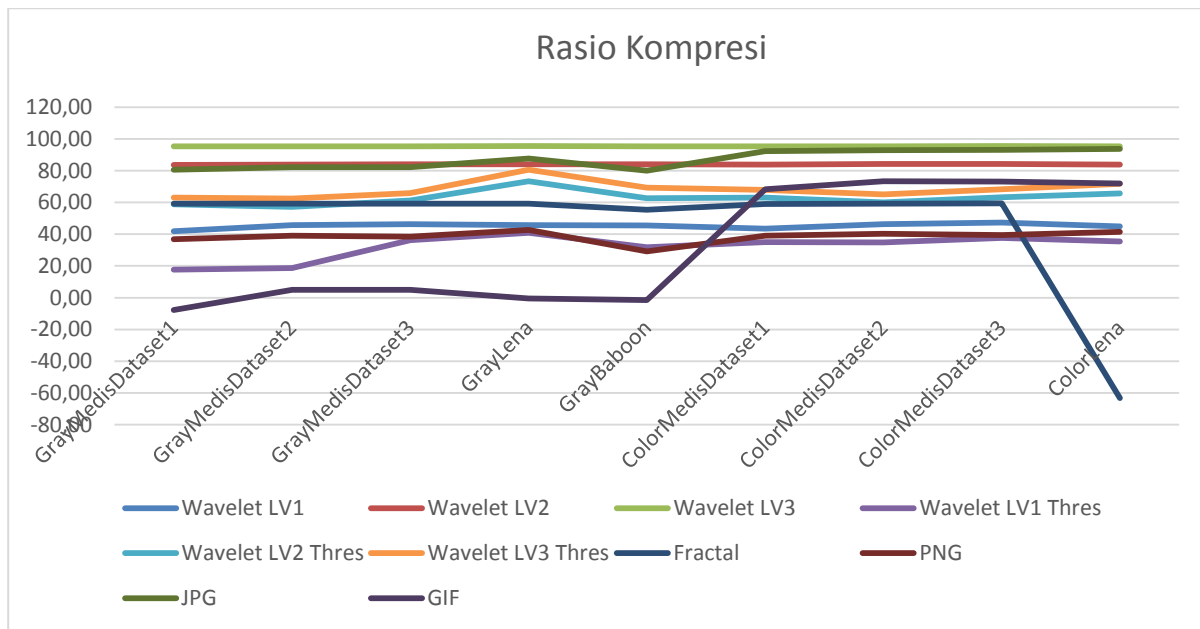
Gambar	Size awal (KB)	Parameter	Wavelet LV1	Wavelet LV2	Wavelet LV3	Wavelet LV1 Thres	Wavelet LV2 Thres	Wavelet LV3 Thres	Fractal	PNG	JPG	GIF
GrayMedis Dataset1	258	PSNR	28.82	23.36	19.85	30.57	25.14	22.53	21.14	inf	37.76	inf
		Size (KB)	149.82	42.42	12.01	212.45	106.16	95.32	105	163	50	278
GrayMedis Dataset2	258	PSNR	27.86	22.66	19.42	29.88	24.86	22.17	21.14	inf	37.01	inf
		Size (KB)	140.07	41.56	11.94	209.87	110.29	96.85	105	157	46	245
GrayMedis Dataset3	258	PSNR	28.91	23.97	20.88	30.7	26.01	23.65	21.28	inf	36.47	inf
		Size (KB)	138.53	41.08	11.88	164.7	100.08	87.77	105	159	46	245
GrayLena	258	PSNR	31.54	26.91	24.66	33.33	29.41	26.54	22.62	inf	37.8	inf
		Size (KB)	140.05	40.97	11.63	152.76	68.87	49.85	105	148	32	259
GrayBaboon	65	PSNR	28.24	26.42	25.32	29.80	28.04	27.26	21.17	inf	31.96	inf
		Size (KB)	35.43	10.31	3.00	44.38	24.30	19.91	29	46	13	66
ColorMedis Dataset1	769	PSNR	29.34	23.93	20.46	31.16	26.39	23.16	22.01	inf	31.43	inf
		Size (KB)	434.37	124.19	35.06	499.51	283.36	247.01	315	469	59	243
ColorMedis Dataset2	769	PSNR	28.08	22.94	19.78	30.11	25.52	22.53	21.19	inf	30.70	inf
		Size (KB)	412.80	121.33	35.00	500.96	307.85	267.75	313	459	54	205
ColorMedis Dataset3	769	PSNR	29.04	24.24	21.30	30.89	26.69	24.07	21.09	inf	30.90	inf
		Size (KB)	405.63	120.27	34.65	479.13	282.49	242.96	312	466	53	207
ColorLena	193	PSNR	29.22	24.80	21.76	31.55	27.6	24.75	22,12	inf	32.96	inf
		Size (KB)	106.48	31.18	8.87	124.74	66.16	54.34	315	113	12	54
ColorBaboon	76	PSNR	23.20	20.66	19.36	25.89	23.45	22.18	22,2	inf	49.54	inf
		Size (KB)	460.93	128.991	35.36	692.10	491.94	452.93	315	588	76	222

Tabel 5.16 Perbandingan Threshold pada Dekomposisi Wavelet LV1

Gambar	Size Awal (KB)	Treshold 2		Treshold 10		Treshold 20		Treshold 30		Treshold 50	
		PSNR	Size Akhir (KB)	PSNR	Size Akhir (KB)	PSNR	Size Akhir (KB)	PSNR	Size Akhir (KB)	PSNR	Size Akhir (KB)
GrayMedisDataset1	258	23.97	20.88	30.7	26.01	23.65	28.91	23.97	20.88	30.7	26.01
GrayMedisDataset2	258	41.08	11.88	164.7	100.08	87.77	138.53	41.08	11.88	164.7	100.08
GrayMedisDataset3	258	23.97	20.88	30.7	26.01	23.65	28.91	23.97	20.88	30.7	26.01
GrayLena	258	41.08	11.88	164.7	100.08	87.77	138.53	41.08	11.88	164.7	100.08
GrayBaboon	65	23.97	20.88	30.7	26.01	23.65	28.91	23.97	20.88	30.7	26.01
ColorMedisDataset1	769	41.08	11.88	164.7	100.08	87.77	138.53	41.08	11.88	164.7	100.08
ColorMedisDataset2	769	23.97	20.88	30.7	26.01	23.65	28.91	23.97	20.88	30.7	26.01
ColorMedisDataset3	769	41.08	11.88	164.7	100.08	87.77	138.53	41.08	11.88	164.7	100.08
ColorLena	193	23.97	20.88	30.7	26.01	23.65	28.91	23.97	20.88	30.7	26.01
ColorBaboon	76	41.08	11.88	164.7	100.08	87.77	138.53	41.08	11.88	164.7	100.08

Tabel 5.17 Perbandingan Persentase Rasio Kompresi

Gambar	Size awal (KB)	Parameter	Wavelet LV1	Wavelet LV2	Wavelet LV3	Wavelet LV1 Thres	Wavelet LV2 Thres	Wavelet LV3 Thres	Fractal	PNG	JPG	GIF
GrayMedis Dataset1	258	Rasio (%)	41.93	83.56	95.34	17.66	58.85	63.05	59.30	36.82	80.62	-7.75
		Size (KB)	149.82	42.42	12.01	212.45	106.16	95.32	105	163	50	278
GrayMedis Dataset2	258	Rasio (%)	45.71	83.89	95.37	18.66	57.25	62.46	59.30	39.15	82.17	5.04
		Size (KB)	140.07	41.56	11.94	209.87	110.29	96.85	105	157	46	245
GrayMedis Dataset3	258	Rasio (%)	46.31	84.08	95.40	36.16	61.21	65.98	59.30	38.37	82.17	5.04
		Size (KB)	138.53	41.08	11.88	164.7	100.08	87.77	105	159	46	245
GrayLena	258	Rasio (%)	45.72	84.12	95.49	40.79	73.31	80.68	59.30	42.64	87.60	-0.39
		Size (KB)	140.05	40.97	11.63	152.76	68.87	49.85	105	148	32	259
GrayBaboon	65	Rasio (%)	45.49	84.14	95.38	31.72	62.62	69.37	55.38	29.23	80.00	-1.54
		Size (KB)	35.43	10.31	3	44.38	24.3	19.91	29	46	13	66
ColorMedis Dataset1	769	Rasio (%)	43.51	83.85	95.44	35.04	63.15	67.88	59.04	39.01	92.33	68.40
		Size (KB)	434.37	124.19	35.06	499.51	283.36	247.01	315	469	59	243
ColorMedis Dataset2	769	Rasio (%)	46.32	84.22	95.45	34.86	59.97	65.18	59.30	40.31	92.98	73.34
		Size (KB)	412.8	121.33	35	500.96	307.85	267.75	313	459	54	205
ColorMedis Dataset3	769	Rasio (%)	47.25	84.36	95.49	37.69	63.27	68.41	59.43	39.40	93.11	73.08
		Size (KB)	405.63	120.27	34.65	479.13	282.49	242.96	312	466	53	207
ColorLena	193	Rasio (%)	44.83	83.84	95.40	35.37	65.72	71.84	-63.21	41.45	93.78	72.02
		Size (KB)	106.48	31.18	8.87	124.74	66.16	54.34	315	113	12	54
ColorBaboon	76	Rasio (%)	-506.49	-69.73	53.47	-810.66	-547.29	-495.96	-314.47	-673.68	0.00	-192.11
		Size (KB)	460.93	128.991	35.36	692.1	491.94	452.93	315	588	76	222



Gambar 5.15 Grafik Persentase Kompresi

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Bab VI ini membahas tentang kesimpulan yang didasari oleh hasil uji coba pada bab sebelumnya. Kesimpulan tersebut nantinya menjawab rumusan masalah yang telah ada pada pendahuluan. Selain itu, juga terdapat saran sebagai acuan untuk mengembangkan topik Tugas Akhir ini lebih lanjut di masa depan.

6.1 Kesimpulan

Dari hasil uji coba yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Dari 7 skenario uji coba yang digunakan, waktu kompresi dari kompresi wavelet jauh lebih cepat dibanding fraktal, dengan waktu kompresi tercepat kurang dari 1 detik. Sedangkan kompresi fraktal yang terlama mencapai 27000 detik.
2. Dari skenario uji coba yang dilakukan ukuran pemampatan atau kompresi citra dengan metode fraktal menghasilkan ukuran file yang bergantung pada besarnya resolusi gambar, sedangkan kompresi wavelet bergantung pada besar kecilnya resolusi serta nilai pixelnya. Dibanding wavelet LV1 kompresi fraktal memiliki *compression ratio* rata-rata diatas 2 dan BPP diatas 4, namun bila dibandingkan dengan wavelet LV 2 dan 3 maka rasio kompresi fraktal masih jauh lebih kecil. Untuk kompresi wavelet LV 3 nilai rasio kompresi mencapai angka diatas 20 dan BPPnya mencapai 0.24. Selain itu nilai threshold juga mempengaruhi nilai PSNR dan ukuran kompresinya.
3. Dari skenario uji coba yang dilakukan, nilai PSNR dan MSE kompresi wavelet jauh lebih baik dibandingkan dengan kompresi fraktal, dengan nilai PSNR tertinggi mencapai 33.33 dan MSE 16.55.

6.2 Saran

Saran yang diberikan untuk pengembangan perangkat lunak ini adalah:

1. Untuk kompresi citra yang diharapkan dalam waktu singkat maka kompresi wavelet merupakan pilihan yang baik..
2. Analisa jenis gambar dapat berpengaruh terhadap pemilihan jenis kompresi.

Daftar Pustaka

- [1] D. Putra, Pengolahan Citra Digital, Yogyakarta: Andi, 2010.
- [2] R. Jasmi, M. Perumal and D. M. Rajasekaran, "Comparison Of Image Compression Techniques Using Huffman Coding, DWT, and Fractal Algorithm," in *IEEE Conference Publications*, Coimbatore, 2015.
- [3] B. M. Ismail, S. M. Basha and B. E. Reddy, "Improved Fractal Image Compression Using Range Block Size," in *IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)*, Bhubaneswar, 2015.
- [4] R. C. Gonzalez and R. E. Wood, Digital Image Processing 3rd ed, New Jersey: Pearson, 2008.
- [5] R. Munir, Pengolahan Citra Digital dengan Pendekatan Algoritmik, Bandung: Informatika, 2004.
- [6] D. Salomon and G. Motta, Handbook of Data Compression 5th ed, Chicago: Springer, 2010.
- [7] S. S. Krivenko, V. V. Lukin and E. Kolganove, "Peculiarities of Lossy Compression of Optical and Medical Images Corrupted by Noise," in *Electronics and Nanotechnology (ELNANO), 2016 IEEE 36th International Conference*, Kiev, 2016.
- [8] S. Madenda, Pengolahan Citra & Video Digital, Jakarta: Erlangga, 2015.
- [9] E. Prasetyo, Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab, Gresik: Andi, 2011.
- [10] M. Samad and A. . A. Suman, "A Reconstruction of Original Image from Destroyed Image through Haar Wavelet Transform," *International Journal of Signal Processing Systems*, vol. 2, no. 2, pp. 60-63, 2014.

- [11] A. Basuki, J. F. Palandi and F. , Pengolahan Citra Digital menggunakan Visual Basic, Yogyakarta: Graha Ilmu, 2005.

BIODATA PENULIS



Nanang Taufan Budiansyah, lahir di Banjarnegara pada tanggal 10 Juli 1995. Penulis menempuh pendidikan mulai dari TK Aisyah Mataram (1999-2001), SDN 2 Cakranegara (2001-2007), SMPN 2 Mataram (2007-2010), SMAN 5 Mataram (2010-2013), dan sekarang sedang menjalani pendidikan S1 Teknik Informatika di ITS. Penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Computer

(HMTC) dan Schematics HMTC. Diantaranya adalah menjadi staff departemen hubungan luar himpunan mahasiswa teknik computer ITS 2014-2015 dan bendahara umum Schematics HMTC ITS 2015-2016. Penulis juga aktif dalam kegiatan kepanitiaan ITS EXPO. Diantaranya penulis pernah menjadi staff Pasar Malam ITS EXPO 2014 dan 2015. Penulis mengambil bidang minat Komputasi Cerdas Visual (KCV). Komunikasi dengan penulis dapat melalui email: **nanangtaufan@gmail.com**.

